But you shall remember *God*

because it is *He* who gives you the ability...

Deuteronomy 8:18

University of Alberta

Dynamic Edge Tracing:
Recursive Methods for Medical Image Segmentation

by

Daniel James Withey

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Medical Sciences – Biomedical Engineering

Department of Electrical and Computer Engineering

Edmonton, Alberta

Spring 2006

Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 0-494-14065-8
Our file Notre référence
ISBN: 0-494-14065-8

NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:
L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

Abstract

Medical image segmentation is a sufficiently complex problem that no single strategy has proven to be completely effective. Historically, region growing, clustering, and edge tracing have been used and while significant steps have been made in the first two, research into automatic, recursive, edge following has not kept pace. In this thesis, a new, advanced, edge tracing strategy based on recursive, target tracking algorithms and suitable for use in segmenting magnetic resonance (MR) and computed tomography (CT) medical images is presented.

This work represents the first application of recursive, target-tracking-based, edge tracing to the segmentation of MR and CT images of the head. Three algorithms representing three stages of development are described. In the third stage, pixel classification data are combined with edge information to guide the formation of the object boundary, and smooth, subpixel-resolution contours are obtained. Results from tests in images containing noise, intensity nonuniformity, and partial volume averaging indicate that the edge tracing algorithm can produce segmentation quality comparable to that from methods based on clustering and active contours, when closed contours can be formed. In addition, low-contrast boundaries can be identified in cases where the other methods may fail, indicating that the information extracted by the edge tracing algorithm is not a subset of that from the other approaches. Additional investigation may allow: 1) the use of knowledge to further guide the segmentation process; and, 2) the formation of multiple segmentation interpretations to be provided as output to the operator or as input to higher-level, automatic processing.

A literature review describing the most common medical image segmentation algorithms is also provided. Three generations of development are defined as a framework for classifying these algorithms.

Acknowledgments

Thanks to my supervisors, Dr. Z. Koles and Dr. W. Pedrycz, for valuable discussions that lent perspective to my initiative. Thanks also to Natasha Kuzbik and Doug Vujanic who worked with early renditions of the mtrack software, and also to Aisha Yahya for her expertise with the surface-display tools.

Financial support from Dr. Koles along with an ample supply of awards and teaching/research assistantships from, or through, the Faculty of Graduate Studies and Research, Province of Alberta, Faculty of Medicine and Dentistry, Department of Biomedical Engineering, and Department of Electrical and Computer Engineering contributed greatly toward the completion of this research.

The consistent support and encouragement of my family and friends was gratefully accepted throughout the course of this program and is gratefully acknowledged. Also, thanks to my colleagues within the EEG group, the BME department, and the ECE department at the University of Alberta for numerous shared thoughts and generous laughter. The students and staff that I had the pleasure to meet truly added another dimension to this experience. The BME soccer team was great.

Certain studies described in this thesis would not have been possible without images and database segmentations from the McConnell Brain Imaging Centre at the Montreal Neurological Institute (available at http://www.bic.mni.mcgill.ca/brainweb/), and the Center for Morphometric Analysis at the Massachusetts General Hospital (available at http://www.cma.mgh.harvard.edu/ibsr/).

Table of Contents

Chapter 1
Introduction **1**
1.1 Motivation 1
1.1.1 EEG Source Localization 1
1.1.2 Realistic Head Models 2
1.2 Medical Image Segmentation 3
1.2.1 Segmentation Problems 4
1.3 Research Direction 5
1.4 Thesis Organization 9
1.5 References 9
Chapter 2
Literature Review **13**
2.1 Segmentation Methods 13
2.1.1 First Generation 15
2.1.1.1 Thresholds 15
2.1.1.2 Region Growing 16
2.1.1.3 Region Split/Merge 16
2.1.1.4 Edge Detection 16
2.1.1.5 Edge Tracing 17
2.1.2 Second Generation 17
2.1.2.1 Statistical Pattern Recognition 18
2.1.2.2 C-means Clustering 20
2.1.2.3 Fuzzy Connectedness 21
2.1.2.4 Deformable Models 22
2.1.2.5 Watershed Algorithm 23
2.1.2.6 Neural Networks 24
2.1.2.7 Multiresolution Methods 26
2.1.2.8 Edge Tracing 27
2.1.3 Third Generation 27
2.1.3.1 Method Combinations 28
2.1.3.2 Knowledge-based Segmentation 29
2.1.3.2.1 Atlas-based Segmentation 29
2.1.3.2.2 Rule-based Segmentation 30
2.1.3.2.3 Model-based Segmentation 32
2.2 Segmentation Software 34
2.2.1 BIC Software Toolbox 35
2.2.2 SPM 35
2.2.3 FSL 35
2.2.4 MEDx 36
2.2.5 EIKONA3D 36
2.2.6 FreeSurfer 36
2.2.7 Insight Segmentation and Registration Toolkit 37
2.2.8 Analyze 37
2.2.9 3D Slicer 38
2.3 References 38

Chapter 3
Dynamic Edge Tracing for 2D Image Segmentation 57
3.1 Introduction 57
3.2 Methodology 59
3.2.1 Synthetic Images 59
3.2.2 Fuzzy c-Means Clustering 60
3.2.3 Dynamic Edge Tracing 61
3.3 Results 66
3.4 Discussion 68
3.5 Conclusions 69
3.6 References 69
Chapter 4
Comparison of Dynamic Edge Tracing and Classical Snakes 71
4.1 Introduction 71
4.2 Methodology 74
4.2.1 Classical Snakes 74
4.2.2 Dynamic Edge Tracing 76
4.2.2.1 Dynamic Systems and Target Tracking 76
4.2.2.2 Application to 2D Edge Tracing 82
4.2.2.2.1 Edge Detection and Feature Extraction 82
4.2.2.2.2 Tracking Algorithm 83
4.3 Results 89
4.3.1 Synthetic MR Image 90
4.3.2 Real MR Image 94
4.3.3 Real CT Image 97
4.3.4 Execution Time 99
4.4 Discussion 100
4.5 Conclusion 103
4.6 References 104
Chapter 5
Dynamic Edge Tracing for Identification of Boundaries in Medical Images 108
5.1 Introduction 108
5.2 Methodology 112
5.2.1 Snake Automated Partitioning (SNAP) 112
5.2.2 FMRIB Automated Segmentation Tool (FAST) 113
5.2.3 Dynamic Edge Tracing (DTC) 114
5.2.3.1 Edge Detection 115
5.2.3.2 Target Tracking 117
5.2.4 Evaluation 129
5.3 Results 130
5.3.1 Parameter Settings 131
5.3.2 Noise and Intensity Nonuniformity 132
5.3.3 Partial Volume Averaging 136
5.3.4 Execution Time 138
5.4 Discussion 138
5.5 Conclusion 141

5.6 Acknowledgment 142
5.7 References 142
Chapter 6
Discussion and Conclusions 147
6.1 Progression of Development 147
6.2 Head Model Formation 152
6.3 The Medical Image Segmentation Problem 153
6.3.1 Knowledge 155
6.3.2 The Segmentation Standard 155
6.3.3 Operator Interaction 156
6.4 The Role of Edge Tracing in Segmentation 157
6.5 Conclusions 158
6.6 Future Work 160
6.7 References 161
Appendix A
mtrack Software Utility 163
A.1 Introduction 163
A.2 Main Panel 164
A.2.1 Slice 166
A.2.2 Data File 166
A.2.3 Edit Mode 166
A.2.3.1 Edges 166
A.2.3.2 Start Points 167
A.2.3.3 Tracks 167
A.2.3.4 Zoom 168
A.2.3.5 Manual 168
A.2.4 Pan 169
A.2.5 View 169
A.2.6 Threshold Control 170
A.2.7 Edge Detection 170
A.2.8 Colours 171
A.2.9 Track Parameters 172
A.2.9.1 Set 172
A.2.9.2 Classify 173
A.2.10 Surfaces 175
A.3 References 179

List of Tables

Table 4-1. Synthetic MR Image Comparison Data..... 91
Table 4-2. Edge Tracing Parameters (Figure 4-6)..... 93
Table 4-3. Edge Tracing Parameters (Figure 4-7e)..... 95
Table 4-4. Edge Tracing Parameters (Figure 4-9)..... 98
Table 5-1 – Hausdorff Distance (pixels) for MNI Slice 95..... 135
Table 5-2 – Metric Comparison for DTC-FAST Combination..... 135
Table 5-3 – Hausdorff Distance (pixels) - IBSR_01 Slices..... 135

List of Figures

Figure 3-1. Synthetic Test Images. 59
Figure 3-2. Tracking System Block Diagram. 63
Figure 3-3. Segmentation Results. 67
Figure 4-1. Tracking System Block Diagram 77
Figure 4-2. Example of Data Association. 80
Figure 4-3. Processing Steps 81
Figure 4-4. Distance Measure. 87
Figure 4-5. Edge Examples. 89
Figure 4-6. Synthetic MR Image. 92
Figure 4-7. Real MR Image. 93
Figure 4-8. Effect of Spatial Dynamics Parameter. 96
Figure 4-9. CT Image - Soft Tissue Boundary. 96
Figure 4-10. Intensity Features. 98
Figure 5-1. Processing Steps 114
Figure 5-2. Edge Features. 116
Figure 5-3. Tracking System Block Diagram 118
Figure 5-4. Intensity Dynamics Example. 121
Figure 5-5. Data Association. 123
Figure 5-6. Threshold Classification 124
Figure 5-7. Use of the Classification Image. 126
Figure 5-8. Similarity Measure. 134
Figure 5-9. IBSR_01 Slice 80. 137
Figure 6-1. MR Segmentation Surfaces 153
Figure A-1. Main Panel and Image Display 165
Figure A-2. Colours Menu 171
Figure A-3. Tracking Parameters 172
Figure A-4. Threshold Classification Menu 173
Figure A-5. Analyze Format Classification Data 175
Figure A-6. Surface Parameters 176
Figure A-7. Surface Display Examples. 179

List of Symbols

$\mathbf{A}$	Dynamic system state transition matrix.
B	Low intensity level adjacent to an edge point in an image.
CSF	Cerebrospinal fluid.
CT	Computed tomography.
DTC	Dynamic edge tracing with pixel classification.
$E\{\cdot\}$	Mathematical expectation.
$f(x,y)$	Two-dimensional image.
$f_{\sigma_c}(x,y)$	Two-dimensional, Gaussian filtered image.
GM	Gray matter.
$\mathbf{H}$	Dynamic system measurement matrix.
k	Sequence counter.
$\mathbf{K}$	Kalman filter gain matrix.
MR	Magnetic resonance.
PD	Proton density.
$\mathbf{P}$	Kalman filter a posteriori estimation error covariance matrix.
$\tilde{\mathbf{P}}$	Kalman filter a priori estimation error covariance matrix.
q	Dynamics parameter. Coefficient of process noise covariance matrix for spatial edge parameters.
q_B	Dynamics parameter. Coefficient of process noise covariance matrix for low intensity edge feature.
q_T	Dynamics parameter. Coefficient of process noise covariance matrix for high intensity edge feature.

$\mathbf{Q}$	Dynamic system process noise covariance matrix.
$\mathbf{R}$	Dynamic system measurement noise covariance matrix.
$\Re$	The set of real numbers.
$\mathbf{S}$	Kalman filter innovation covariance matrix.
SNR	Signal to noise ratio.
T	High intensity level adjacent to an edge point in an image.
$\mathbf{v}$	Dynamic system measurement noise vector.
$\mathbf{w}$	Dynamic system process noise vector.
WM	White matter.
$\mathbf{x}$	Dynamic system state vector.
$\hat{\mathbf{x}}$	Dynamic system state vector estimate.
$\tilde{\mathbf{x}}$	Dynamic system state vector prediction.
$\mathbf{y}$	Kalman filter innovation vector.
$\mathbf{z}$	Dynamic system measurement vector.
Δ	Time step.
σ_c	Standard deviation for Gaussian filter.
σ_N^2	Signal noise variance.

Chapter 1

Introduction

1.1 Motivation

1.1.1 EEG Source Localization

Epilepsy is a neurological disorder that affects 0.5% to 2% of the North American population [1], [2]. New cases are found most frequently in individuals under the age of 10 and those over the age of 60 [1], [2]. The disease is characterized by seizures, sudden episodes of uncontrolled, neural activity that may vary in severity and frequency from patient to patient.

An electroencephalogram (EEG) is a recording of voltage versus time from a set of electrodes placed on the scalp. It is known that these voltage measurements reflect underlying activity in the brain [3]. In epilepsy, abnormal neural activity occurs which is often manifested in the EEG. This information is used clinically for determining

diagnosis and treatment but its impact is usually limited to a qualitative interpretation by a neurologist.

Mathematical techniques can be used to analyze the EEG [4] with the goal of accurately locating the source of abnormal activity within the brain. This is most effective when the patient's seizures are of a type classified as partial, meaning that they arise from a focal point within the brain, including those with secondary generalization. Approximately 60% of adult epilepsy patients experience partial seizures [1]. Accuracy of source localization is very important when surgery is a treatment option but knowledge of the source location can also aid in the selection of medication.

1.1.2 Realistic Head Models

Mathematical EEG analysis requires a model describing the spatial distribution of electrical conductivity within the head. This permits seizure information in the EEG voltage measurements to be projected back inside the head, in the model, to identify possible source locations. A model using a spherical head approximation has often been used but it has been recognized that models based on the patient's own anatomy improve the accuracy of the localization [5]-[7].

Three-dimensional (3D) anatomical information can be obtained from medical imaging techniques that provide information on tissue structure, namely, magnetic resonance (MR) imaging and X-ray computed tomography (CT). The X-ray CT images show bone very well and MR images are very good for soft tissue discrimination. Segmentation of these images into component tissue volumes provides a basis for obtaining realistically-shaped, patient-specific, electrical, head models [6], [8], [9].

Other medical imaging techniques such as positron emission tomography (PET), single photon emission computed tomography (SPECT), and functional magnetic resonance imaging (fMRI) provide information regarding tissue function. These are less useful than structural information for the development of electrical head models and are not typically used for that purpose.

In cases where MR and CT images are not available, realistic head models have been formed from a generic surface model containing scalp, skull, and brain surfaces, deformed to match a set of points measured on the patient's scalp. It is recognized, though, that this is less accurate than forming the head model from segmented images [10].

1.2 Medical Image Segmentation

Medical images are typically held as two-dimensional (2D) arrays of picture elements (pixels) or three-dimensional (3D) arrays of volume elements (voxels, also called pixels). Segmentation is the process of separating these images into component parts. Specifically, scalp, skull, gray matter, white matter, and cerebrospinal fluid are important tissue classes for the formation of electrical head models. Segmentation can be performed by the identification of a surface for each tissue class, or by the classification of each pixel in the image volume.

Manual segmentation of CT and MR images is possible but it is a time consuming task and is subject to operator variability. Therefore, reproducing a manual segmentation result is difficult and the level of confidence ascribed to it may suffer accordingly. For these reasons, automatic methods are considered to be preferable [11]; however,

significant problems must be overcome to perform segmentation by automatic means and it remains an active research area.

1.2.1 Segmentation Problems

Segmentation of medical images involves three main image related problems. The images may contain noise that can alter the intensity of a pixel such that its classification becomes uncertain. Also, the images can contain intensity nonuniformity where the average intensity level of a single tissue class varies over the extent of the image. Third, the images have finite pixel size and are subject to partial volume averaging where individual pixels contain a mixture of tissue classes and the intensity of a pixel may not be consistent with any single tissue class.

These image-related problems and the variability in tissue distribution among individuals in the human population leaves some degree of uncertainty attached to all segmentation results. This includes segmentations performed by medical experts where variability occurs between experts (inter-expert variability) as well as for a given expert performing the same segmentation on multiple occasions (intra-expert variability). Despite this variability, image interpretation by medical experts must still be considered to be the only available truth for in vivo imaging [11].

Medical image segmentation must, therefore, be classed as an underdetermined problem where the known information is not sufficient to allow the identification of a unique solution. The challenge in developing automatic segmentation methods is in the selection of mathematical models, algorithms, and related parameter values to compensate for the missing information and produce a solution that falls within a set of

acceptable solutions, that is, within the spatial limits of the inter- and intra-expert variability. So far, this has not been achieved in a way that permits general application.

The use of automatic methods requires evaluation against a truth model to obtain a quantitative measurement of the efficacy of a given algorithm. Evaluation of results from automatic segmentation of in vivo images is usually accomplished by comparison with segmentations made by experts. Additional evaluation of an algorithm is possible by the analysis of synthetic images or images of physical phantoms [12].

A final problem occurs when an automatic method is employed for a segmentation task and the result is deemed to be unacceptable by the operator. This problem is not often considered by those interested solely in algorithmic detail; however, faulty segmentations must be corrected to have clinical usefulness. Modifying unacceptable, automatically-generated results is a process that may require hours of tedious manual effort.

1.3 Research Direction

Despite much effort by researchers in many countries, automatic medical image segmentation remains an unsolved problem, making the development of new algorithms important. The underdetermined nature of the problem and the experience of past research suggest that the use of uncertainty models, optimization methods, and the ability to combine information from diverse sources are important characteristics.

An examination of algorithms that existed at the beginning of this research program suggested that those which used boundary information were unable to use image region information well and those that used region information did not use boundary information very well. Acquiring both boundary and region information appeared to be important for

producing suitable segmentations, a conclusion that has also been drawn by others [13], [14].

An important conceptual generalization suggests that algorithms that use primarily local information are not able to incorporate global information easily and those that use global information well have difficulty incorporating local information. For example, the pixel classification methods, such as clustering, classify pixels drawn from across the entire image but completely lose the local context of a given pixel unless special attention is applied. On the other hand, the deformable models produce object boundaries by many local deformations but may not find the desired boundary at all points.

It was also recognized that an analogy exists between edge tracing, the propagation of a contour along an edge, and target tracking algorithms used in the military/aerospace industry for tracking maneuvering targets, often in adverse conditions where measurement information may be corrupted by noise and nearby objects. Target tracking algorithms [15]-[17] utilize uncertainty models, optimization methods and are capable of combining diverse pieces of information, precisely the characteristics needed for image segmentation. Given this apparent match of capability to requirement, the hypothesis was formed that target tracking algorithms could be used for the foundation of a new image segmentation strategy capable of combining local and global information to form contours automatically around objects in medical images.

The resulting investigation produced the concept of dynamic edge tracing, a new approach to image segmentation suitable for MR and CT images where a dynamic system model is used to interpret edge information and statistically-based, target tracking algorithms automatically associate edge points into object boundaries.

Edge tracing may initially be viewed as an unlikely candidate for a successful segmentation strategy. Although it is one of the earliest segmentation methods [18] and is conceptually similar to segmentation operations performed by human experts, it is among the least researched at present and is not highly regarded in the image analysis community where poor robustness has led researchers to disregard it in favour of other methods [11], [13]. In fact, research into automatic, recursive, edge-based methods has largely been lost during the development of segmentation algorithms over the past two decades and presently little or no representation is found in major review articles [11], [12].

The criticism that has been leveled at edge tracing algorithms includes: i) sensitivity to noise; ii) the potential for gaps in the boundaries that are formed; and iii) the potential for false edges to be included in the boundary [13]. These have the combined effect of producing low robustness in the segmentation process.

What appears to go unrecognized is that the identification of a coherent boundary by linking neighbouring edge points provides useful information for the purpose of segmentation, information not obtained by other methods. This is particularly evident along low-contrast boundaries. Furthermore, edge tracing based on target tracking has the ability to combine, or fuse, a wide variety of information including results from other algorithms.

Related, previous work [19], [20], has not exploited the potential of this technique, focusing on tracking in a single spatial dimension, and would not be applicable to the segmentation of MR and CT head images where the identification of convoluted, nonconvex contours is required.

Dynamic edge tracing is capable of incorporating both local and global information by combining edge, intensity and pixel classification data, to identify object boundaries in medical images. Unlike other edge tracing methods, this approach has no restrictions related to object smoothness or convexity and appears to be the first target-tracking-based, edge tracing algorithm to be applied to the segmentation of MR and CT head images. When closed contours can be formed, it can produce segmentations comparable to those from other algorithms over a range of conditions involving noise, intensity nonuniformity, and partial volume averaging.

Dynamic edge tracing is also easily modified or expanded to include additional information. This flexibility facilitates further development and is important because the potential of target tracking algorithms for image segmentation has not yet been fully explored. For example, due to the existence of an array of possible neighbour points that are identified at each step of the tracing process, multiple sets of segmentation interpretations, multiple hypotheses, can be identified. This could produce a much richer set of candidate segmentations than is possible with methods that attempt to find a single solution. These, or a select subset, could then be presented to the operator for evaluation or to higher levels of processing. Algorithms that generate and process multiple hypotheses exist in the target tracking literature [15] but adaptation is required to apply them to the problem of automatic image segmentation. In addition to this, there are ways to utilize domain knowledge to improve the tracing result, for example, in the analysis and selection of neighbour points.

1.4 Thesis Organization

The remainder of this thesis has the following components. Chapter 2 is a brief overview of past and present medical image segmentation research. The emphasis is on providing a representative summary of major segmentation methods with an adequate supply of references for further investigation. Three generations of development are defined as a framework for classifying the many segmentation methods that have been developed. Chapters 3, 4, and 5 contain studies on the proposed dynamic edge tracing algorithm and represent a progression in its development. Chapter 3, published as [21], is the earliest study and probes the feasibility of dynamic edge tracing using synthetic images containing intensity nonuniformity. Chapter 4 describes a substantially modified algorithm operating on synthetic and real images and with comparison to the classical snakes algorithm, one of the earliest of the now very popular deformable models. Chapter 5 [22] presents further developments of the dynamic edge tracing algorithm with improvements in contour smoothness and incorporation of global image information. Images from a synthetic image database as well as real images with manually determined contours are used for evaluation. Comparison is made with a well known statistical classification method and a region competition, level set method. Chapter 6 provides discussion, conclusions, and ideas for future work. Finally, a description of the software developed to support these investigations is provided in an appendix.

1.5 References

[1] http://www.epilepsy.ca

[2] http://www.epilepsy.com

[3] F. Lopes da Silva, "Neural mechanisms underlying brain waves: from neural membranes to networks," *Electroencephalography and Clinical Neurophysiology*, Vol. 79, 1991, pp. 81-93.

[4] R.D. Pascual-Marqui, "Review of methods for solving the EEG inverse problem," *International Journal of Bioelectromagnetism*, Vol. 1, No. 1, 1999, pp. 75-86.

[5] B.J. Roth, M. Balish, A. Gorbach, S. Sato, "How well does a three-sphere model predict positions of dipoles in a realistically shaped head?," *Electroencephalography and Clinical Neurophysiology*, Vol. 87, 1993, pp. 175-184.

[6] B.N. Cuffin, "EEG localization accuracy improvements using realistically shaped head models," *IEEE Transactions on Biomedical Engineering*, Vol. 43, No. 3, 1996, pp. 299-303.

[7] G. Huiskamp, M. Vroeijenstijn, R. van Dijk, G. Wieneke, A.C. van Huffelen, "The need for correct realistic geometry in the inverse EEG problem," *IEEE Transactions on Biomedical Engineering*, Vol. 46, No. 11, 1999, pp. 1281-1287.

[8] T. Heinonen, H. Eskola, P. Dastidar, P. Laarne, J. Malmivuo, "Segmentation of T1 MR scans for reconstruction of resistive head models," *Computer Methods and Programs in Biomedicine*, Vol. 54, 1997, pp. 173-181.

[9] H.J. Wieringa, M.J. Peters, "Processing MRI data for electromagnetic source imaging," *Medical and Biological Engineering and Computing*, Vol. 31, 1993, pp. 600-606.

[10] J. Koikkalainen, J. Lötjönen, "Reconstruction of 3-D head geometry from digitized point sets: An evaluation study," *IEEE Transactions on Information Technology in Biomedicine*, Vol. 8, No. 3, 2004, pp. 377-386.

[11] L.P. Clarke, R.P. Velthuizen, M.A. Camacho, J.J. Heine, M. Vaidyanathan, L.O. Hall, R.W. Thatcher, M.L. Silbiger, "MRI segmentation: Methods and applications," *Magnetic Resonance Imaging*, Vol. 13, No. 3, 1995, pp. 343-368.

[12] D.L. Pham, C. Xu, J.L. Prince, "Current methods in medical image segmentation," *Annual Review of Biomedical Engineering*, Vol. 2, 2000, pp. 315-337.

[13] J.S. Suri, S. Singh, L. Reden, "Computer vision and pattern recognition techniques for 2-D and 3-D MR cerebral cortical segmentation (Part 1): A state-of-the-art review," *Pattern Analysis and Applications*, Vol. 5, 2002, pp.46-76.

[14] J.S. Suri, S. Singh, L. Reden, "Fusion of region and boundary/surface-based computer vision and pattern recognition techniques for 2-D and 3-D MR cerebral cortical segmentation (Part 2): A state-of-the-art review," *Pattern Analysis and Applications*, Vol. 5, 2002, pp.77-98.

[15] S. Blackman, R. Popoli, "Design and Analysis of Modern Tracking Systems," Artech House, 1999.

[16] E. Waltz, J. Llinas, "Multisensor Data Fusion", Artech House, 1990.

[17] Y. Bar-Shalom, T.E. Fortmann, "Tracking and Data Association," Academic Press, 1988.

[18] K.S. Fu and J.K. Mui, "A survey on image segmentation," *Pattern Recognition*, Vol. 13, 1981, pp. 3-16.

[19] M. Basseville, B. Espiau, J. Gasnier, "Edge detection using sequential methods for change in level – Part I: A sequential edge detection algorithm," *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. ASSP-29, No. 1, 1981, pp. 24-31.

[20] P. Abolmaesumi, M.R. Sirouspour, "An interacting multiple model probabilistic data association filter for cavity boundary extraction from ultrasound images," *IEEE Transactions on Medical Imaging*, Vol. 23, No. 6, 2004, pp. 772-784.

[21] D.J. Withey, Z.J. Koles, W. Pedrycz, "Dynamic edge tracing for 2D image segmentation," in: Proc. 23rd Int. Conf. IEEE Engineering in Medicine and Biology Society, Vol. 3, Oct. 2001, pp. 2657-2660.

[22] D.J. Withey, W. Pedrycz, Z.J. Koles, "Dynamic edge tracing for identification of boundaries in medical images," submitted to *Computer Vision and Image Understanding*, 2005.

Chapter 2

Literature Review

2.1 Segmentation Methods

Automatic segmentation methods have been classified as either supervised or unsupervised [1]. Supervised segmentation requires operator interaction throughout the segmentation process whereas unsupervised methods generally require operator involvement only after the segmentation is complete. Unsupervised methods are preferred to ensure a reproducible result [2]; however, operator interaction is still required for error correction in the event of an inadequate result [3].

Objects within 2D or 3D images can be identified either by labeling all pixels in the object volume, or by identifying boundaries of the objects. Some segmentation methods may also be categorized in this manner, as volume identification methods or as boundary identification methods. In the volume identification type, each pixel is assigned a label from which object boundaries may subsequently be derived. The complement, boundary

identification, consists of techniques that initially identify object boundaries, from which the labeling of pixels within the boundaries may follow.

When considering the image segmentation literature it should be noted that there are subtle distinctions in application that may not be discernible from the title of a particular publication. For example, "segmentation of the brain" may refer to the extraction of the whole brain volume, which is a somewhat different problem than that of attempting to differentiate between tissue regions within the brain. Also, some segmentation methods are only intended to operate on the brain image after the skull and scalp have been removed. Automatic segmentation of full head images, those including brain and scalp, is more complicated because intensity levels from the scalp often overlap those from the brain.

Most publications concern segmentation of MR images as opposed to CT images. This is probably because more soft tissue detail is possible with MR. In addition, more data are available from MR imaging since multispectral images with different relative tissue intensity levels can be obtained in a single acquisition session. Multispectral images are often used in segmentation methods based on clustering or other pattern recognition techniques, for example.

It is convenient to classify the image segmentation literature into three generations, each representing a new level of algorithmic development. The earliest and lowest level processing methods occupy the first generation. The second is composed of algorithms using image models, optimization methods, and uncertainty models, and the third is characterized by algorithms that are capable of incorporating knowledge. The second generation followed the first chronologically as computing power increased, whereas the

third has begun in parallel with the second, often utilizing methods from the first and second generations.

The number of publications regarding medical image segmentation is quite large and as a result the following information is intended to be representative rather than exhaustive. Review articles [1]-[12] and references cited in the text are sources for related articles and additional details.

2.1.1 First Generation

First-generation techniques can be utilized in supervised or unsupervised segmentation systems but should be considered as low-level techniques since little, if any, prior information is included. They are usually described at a conceptual level leaving the details (e.g. threshold levels, homogeneity criterion) to be determined by the user, often resulting in ad hoc implementations. Relatively simple methods like these are subject to all three of the main image segmentation problems. Further description can be found in textbooks on image processing, for example, [13]-[16].

2.1.1.1 Thresholds

In the simplest case, a threshold can be applied to an image to distinguish regions of different intensity and thus differentiate between classes of objects within the image. Thresholds may also be applied in a higher-dimensional, feature space where better separation of classes may be possible.

Thresholds can be operator-selected or automatically-determined, for example, using information from image gray level histograms. In images where intensity nonuniformity and noise are present it may be difficult or impossible to find one or more thresholds which separate the image objects without misclassification. Still, the application of

thresholds is extremely simple and they continue to be used when the nature of the problem permits or when augmented by additional processing steps [17], [18].

2.1.1.2 Region Growing

Starting at a seed location in the image, adjacent pixels are checked against a predefined homogeneity criterion. Pixels that meet the criterion are included in the region. Continuous application of this rule allows the region to grow, defining the volume of an object in the image by identification of similar, connected pixels.

Region growing continues to be used where the nature of the problem permits [14] and developments continue to be reported [19]-[21].

2.1.1.3 Region Split/Merge

The region split/merge segmentation algorithm [14] operates on an image in a recursive fashion. Beginning with the entire image, a check is performed for homogeneity of pixel intensities. If it is determined that the pixels are not all of similar intensity then the region is split into equal-sized subsections. For 3D images, the volume is split into octants (quadrants for 2D images) and the algorithm is repeated on each of the subsections down to the individual pixel level. This usually results in over-segmentation where homogeneous regions in the original image are represented by a large number of smaller subregions of varying size. A merge step is then performed to aggregate adjacent subregions that have similar intensity levels.

2.1.1.4 Edge Detection

Edge-based methods attempt to describe an object in terms of its bounding contour or surface rather than by the volume that it occupies. Many edge detection operators exist, such as, Sobel, Prewitt, Canny, and wavelet [6] as well as Marr-Hildreth [14]. Some, such

as Sobel and Prewitt, are quite simple and can be implemented by n-linear convolution operations for n-dimensional images. Often this is followed by a computation of the magnitude of the gradient at each pixel position.

Edge detection is typically not suitable for image segmentation on its own since the edges found by application of low-level operators are based on local intensity variations and are not necessarily well connected to form closed boundaries [6], [14]. Therefore, edge detection is often used to supplement other segmentation techniques.

2.1.1.5 Edge Tracing

Edge tracing is a boundary identification method where edge detection is performed to form an edge image after which edge pixels with adjacent neighbour connectivity are followed sequentially and collected into a list to represent an object boundary [13], [22], [23]. Evaluation of a cost function involving a variety of local and global image features is performed in a heuristic search for neighbouring pixels. Unfortunately, these algorithms tend to be very sensitive to noise that creates gaps or diversions in the object boundary. Methods for extracting 3D surfaces, by stacking 2D contours [24] and by a 3D edge following procedure [25], have also been developed.

2.1.2 Second Generation

Research in automatic image segmentation diverges from the first-generation algorithms with the introduction of uncertainty models and optimization methods as well as a general avoidance of heuristics. Segmentation methods can often still be identified as being primarily one of either volume identification or boundary identification and as either supervised or unsupervised.

2.1.2.1 Statistical Pattern Recognition

Statistical pattern recognition [1], [7] has been applied extensively in medical image segmentation. A mixture model is used where each of the pixels in an image is modeled as belonging to one of a known set of classes. For head images, these will be tissue classes such as gray matter, white matter, and cerebrospinal fluid. A set of features, often involving pixel intensity, is evaluated for each pixel. This forms a set of patterns, one for each pixel, and the classification of these patterns assigns probability measures for the inclusion of each pixel in each class.

As part of the process, class conditional probability distributions describing the variation of each pixel feature are often required for each class. These are generally not known and can be determined manually or automatically. For example, in supervised, statistical classification these distributions can be calculated from operator-selected regions acquired from each tissue class in the image. Alternatively, in unsupervised, statistical clustering, the distributions are automatically estimated from the image data, usually requiring an iterative procedure. Not all statistical pattern recognition methods estimate class conditional distributions. Some perform the segmentation directly by cost-function optimization.

Parametric approaches in statistical pattern recognition are those where the forms of the class conditional distributions are known, as, for example, when Gaussian distributions are assumed. Alternatively, nonparametric approaches are those where the forms of the class conditional distributions are not known.

The total number of classes present in the image and the a priori probability of occurrence of each class within the image are assumed to be known prior to the

segmentation operation. For each pixel in the input image, the a posteriori probability that the pixel belongs to each tissue class is generally computed using Bayes' rule [1] and a maximum a posteriori (MAP) rule is applied, where the pixel is assigned to the class in which its a posteriori probability is greatest, to complete the segmentation.

Bayesian classifiers, discriminant analysis, and k-Nearest Neighbour classification are examples of supervised methods that have been applied [26].

Recent research has been performed in the area of unsupervised, volume identification using parametric, statistical clustering implemented with expectation maximization (EM), a two-step, iterative procedure, and where a mixture of Gaussians is assumed for the pixel intensity data. This has allowed segmentation and nonuniformity gain field estimation to occur simultaneously [27]-[29], addressing the intensity nonuniformity problem. The application of a Markov random field (MRF) [30] to introduce contextual information by allowing neighbour pixels to influence classification and by modeling a priori information regarding the possible neighbours for each tissue class, has helped to reduce misclassification errors arising from noise and partial volume averaging [28], [29]. An extension to further address the partial volume problem is found in [31] and a generalization of the EM-MRF approach which uses a hidden Markov random field and EM is reported in [32]. A segmentation method using a variant of the EM algorithm and which estimates a separate bias field for each tissue class is described in [33]. The relatively high computational cost of the EM approach, though, has spurred the search for speed enhancements [34] and alternatives [35].

Statistical models to describe partial volume averaging have been developed, for example [36] and also [37] where a statistical representation for the volume of the segmented object is also computed.

2.1.2.2 C-means Clustering

C-means cluster analysis [1] permits image pixels to be grouped together based on a set of descriptive features. For example, pixel intensity could be used as a feature, causing pixels to be grouped according to intensity levels. Other features which describe individual pixels (e.g. the texture of the local neighbourhood) can also be used to improve cluster separation. The numerical value of each feature is generally normalized to between 0 and 1.

C-means cluster analysis operates in the p-dimensional feature space, where p is the number of features used. Each pixel produces one point in the feature space and a cluster is a region in the feature space having a high density of such points. For each cluster, a cluster centre, or prototype, can be defined. The membership of a pixel in a particular cluster depends on the distance between its feature-space representation and the cluster prototypes.

The number of clusters (c) is assumed to be known. Equations for iterative computation of the positions of the cluster prototypes and the memberships of each pixel in a cluster are determined by minimizing an objective function based on the sum of the distances (i.e. some measure of similarity) between the cluster prototypes and each of the p-dimensional data points. During algorithm operation, there is no guarantee that a global minimum will be attained. The algorithm execution is terminated when the first local minimum is reached.

Hard c-means algorithms assign to each pixel absolute membership in one of the clusters whereas fuzzy c-means algorithms assign to each pixel a degree of membership within each of the clusters. Hardening of the fuzzy result is often done by assigning each pixel to the cluster in which it has highest membership.

Recent research has been performed using adaptive methods based on fuzzy c-means clustering (FCM) for unsupervised, volume identification [38]. The adaptive technique is implemented by modifying the FCM objective function and provides compensation for the intensity nonuniformity problem. Alternatives that reduce computational complexity and add spatial constraints, for reduction of errors due to noise, have also been reported [39]-[41].

2.1.2.3 Fuzzy Connectedness

Fuzzy representations of connectedness between the pixels comprising an object in an image, drawn from early work on fuzzy image analysis by Rosenfeld [42], [43], have been developed for use in medical image segmentation [44], [45]. Udupa and Saha [45] describe several algorithms. Given a seed pixel within an object in an image, the object containing the seed is determined by computing a connectedness measure for all pixels in the image relative to the seed pixel. Final object selection is performed using a threshold on the resulting fuzzy connectedness map. When multiple objects are considered, a seed pixel is required for each object and the fuzzy connectedness of all image pixels to each seed are computed. Pixels are then assigned to the object of highest connectedness. Intensive computation may be required because connectedness is defined based on an optimal path to the seed pixel. Dynamic programming is used to determine optimal paths. Fuzzy connectedness has been used in intra-operative tumour segmentation where a

rectangular, operator-selected region of interest surrounding the tumour has also been applied to reduce computation time [46].

2.1.2.4 Deformable Models

Deformable models, including active contours (2D) and active surfaces (3D), are artificial, closed contours/surfaces able to expand or contract over time, within an image, and conform to specific image features.

One of the earliest active contours is the snake [47], used for supervised, boundary identification in 2D images. The snake is endowed with physical elasticity and rigidity features and intensity gradients in the image are used to derive external forces acting on the snake. During iterative update of an energy-minimization evolution equation, the snake moves to the nearest edge and is able to conform to it, identifying the boundary of an object within the image.

In the early stages of development, the snake needed to be initialized very near to the boundary of interest, had difficulty entering narrow concavities, and had problems discriminating between closely spaced objects. Attempts to overcome these problems resulted in many modifications [9]. Extensions to allow 3D volume segmentation were also developed as was the ability to change topology to handle objects with bifurcations or internal holes [9], [48]. New snake models continue to be developed [49]-[51].

Level set methods were introduced to deformable models by casting the curve evolution problem in terms of front propagation rather than energy minimization [52]-[55]. With level sets, the contour or surface moves in the direction of its normal vectors. The speed of the contour is an important component for maintaining consistent contour propagation and for halting at regions of high gradient. Local contour curvature, intensity

gradient, shape, and contour position can be used in the speed term although the selection need not be limited to these [55]. The development of the level set approach simplified topology adaptation so that a contour or surface could split and merge as it evolved, allowing it to identify boundaries of complex objects. Efforts have also been made to reduce the computational burden [56].

Mumford-Shah segmentation techniques [57], rather than intensity gradient, have been used to form the stopping condition [58] producing a region-based, active contour and this has been further developed to produce a deformable model that finds multiple object boundaries with simultaneous image smoothing [59]. Mumford-Shah segmentation assumes a piecewise smooth image representation and defines a problem in variational calculus where the solution produces simultaneous smoothing and boundary identification in an image [57].

Most deformable models propagate toward a local optimum. A recent, related method for finding globally optimal surfaces by simulating an ideal fluid flow under image-derived, velocity constraints is described in [60].

2.1.2.5 Watershed Algorithm

The watershed algorithm is a boundary identification method in which gray level images are modeled as topographic reliefs where the intensity of a pixel is analogous to the elevation at that point [61]. In a real landscape, catchment basins, e.g. lakes and oceans, are regions each associated with a local minimum. In a similar way, a gray level image has local minima. The watershed concept can be understood by imagining that a hole is cut at each local minimum in the relief and then the relief is immersed, minima first, into water. As the relief is immersed, water rises from the holes in the local minima.

At each point where water would flow from one catchment basin to another, a "dam" is constructed by marking those points. When the entire relief has been immersed in water, the "dams" ring each catchment basin in the image, identifying the boundaries of the local minima. The tendency is to oversegment the image since every local minimum will be identified including those resulting from noise. Thresholds are generally used to suppress shallow minima.

Often edge detection is used to produce a gradient magnitude image for input to the watershed algorithm since the catchment basins will then be the objects of interest, that is, regions not associated with edges in the image.

The watershed algorithm has been used to segment the cerebellum from 3D MR images of the mouse head [62], for example.

2.1.2.6 Neural Networks

Artificial neural networks have been used in medical image segmentation [1], typically in unsupervised, volume identification but also in boundary identification [63]. The network must first be trained with suitable image data, after which it can be used to segment other images. For volume identification, the neural network acts as a classifier where a set of features is determined for each image pixel and presented as input to the neural network. The network uses this input to select the pixel classification from a predefined set of possible classes, based on its training data. The classification operation is like that performed in statistical pattern recognition and it has been noted that many neural network models have an implicit equivalence to a corresponding statistical pattern recognition method [7].

Recent investigations considering biological neurons in animal models have shown that neurons of the visual cortex produce stimulus-dependent synchronization [64]. This has led to the suggestion that the synchronous activity is part of the scene segmentation process. Neural networks have been formed using artificial neurons derived, with significant simplification, from the physiological models and used for unsupervised, volume identification. Examples are pulse coupled neural networks (PCNNs) [65] and the locally excitatory globally inhibitory oscillator network (LEGION) [66]. Neurons are usually arranged in a one-to-one correspondence to the image pixels and have linkages to a neighbourhood of surrounding neurons. Each neuron produces a temporal pulse pattern that depends on the pixel intensity at its input and also on the local coupling. The linkages between neurons permit firing synchrony and the time signal from a group of neurons driven by the same object in an image is specific to that object. The local coupling helps to overcome intensity nonuniformity and noise. Implementations of PCNNs as hardware arrays are being explored with the intent of producing real-time, image-processing systems [65].

Unsupervised, volume identification has also been performed by a method utilizing vector quantization and a deformable feature map where training required one manually segmented dataset [67].

Neural networks have also been used as an autoassociative memory to identify lesions in MR, head images [68]. The network is trained using images from normal subjects. When an image containing an abnormality is presented to the network, the abnormality is recognized as different from the training images.

Neuro-fuzzy systems, combinations of neural networks and fuzzy systems, have also been used in image segmentation. Boskovitz and Guterman [69] provide a brief survey and propose a system which performs image segmentation by neural-network-controlled, adaptive thresholds applied to a "fuzzified" version of the input image obtained by fuzzy clustering.

2.1.2.7 Multiresolution Methods

Multiresolution, multiscale, and pyramid analysis are terms referring to the use of scale reduction to group pixels into image objects. These methods are typically used for unsupervised, volume identification but have also been used in unsupervised, boundary identification. The segmentation is performed by first forming a set, or stack, of images by recursively reducing the scale of the original image by blurring followed by down sampling. The result is a sequence of images that if stacked one above the other from highest resolution to lowest resolution would form a pyramid of images, each determined from the one below. The lowest resolution image (apex of the pyramid) may be as small as 2x2x2 pixels, for 3D images, and the highest resolution image (base of the pyramid) is the original. The pixels are then linked from one layer to the next by comparing similarity attributes, such as intensity features. Pixels that have similar features and location are labeled as belonging to the same object, completing the segmentation.

Simple edge tracing methods have been augmented by further processing using multiresolution pyramids to connect edge discontinuities [70] and boundaries have been refined using a multiscale approach [71]. Examples of volume identification using multiresolution pyramids can be found in [72], [73].

2.1.2.8 Edge Tracing

Optimization methods have been applied to edge following in 2D images using graph search methods [74], [75] for supervised, boundary identification. An initial seed pixel and a destination pixel are specified and the cost for transfer between pixels in the intervening trellis is computed based on a given cost function. The minimum cost path between the initial pixel and the destination pixel can then be computed and the pixel sequence collected into a list. Repetition of this procedure where each destination pixel is used as the initial pixel in a subsequent operation allows the user to guide the formation of an object boundary. Dynamic programming is used for determining the minimum-cost path [75]. An extension to permit user-guided, 3D boundary formation has also been developed [76].

User-guided edge following in 2D images has also been performed by computing geodesic contours [77], as used in deformable models based on the methods of level sets. A geodesic contour is defined as the shortest line on a surface between two points on the surface. Paths between a series of user-defined, boundary points are determined by minimizing a weighted distance measure that is lowest along high-intensity gradients in the image. Computation involves the numerical solution of partial differential equations.

2.1.3 Third Generation

It is important to recognize that medical image segmentation is a challenge even for human experts who have knowledge, training, and awareness at a level that has not been replicated in computer programs. Furthermore, trained experts may disagree on details regarding a particular segmentation [2]. This suggests very strongly that the problem of fully-automatic image segmentation will not be solved by a second-generation algorithm

and that while optimization methods and uncertainty models are important and should be used, they are not sufficient in themselves to produce accurate, automatic segmentations, in the general case. Recent work has led to the development of techniques that combine multiple methods and also to the development of methods that incorporate higher-level knowledge such as a priori information, expert-defined rules, or models (e.g. shape) of the desired object. These constitute the third generation of medical image segmentation methods.

2.1.3.1 Method Combinations

Examples of combining second-generation, volume identification with second–generation, boundary identification are: multiresolution concepts with watershed algorithm [78], neural network classification and active contours [79], fuzzy clustering and an active contour method [80], statistical methods and active contours [81]-[83].

An approach based on information theory was used to combine pixel classification, edge detection, and shape constraints in [84].

The FreeSurfer software [85] performs cortical surface reconstruction from T1-weighted, MR, head images using a multistep procedure that includes the use of deformable models, watershed algorithm, and a variety of signal processing techniques (e.g. smoothing, histogram analysis) [86]-[89]. Another approach for identification of the cerebral cortex utilizes the adaptive fuzzy clustering of [38], deformable models, region growing, and additional signal processing [90]. A third approach to cortical surface formation uses a nonparametric classifier, a deformable model, intensity nonuniformity correction, and includes a statistical model for partial volume averaging [91]. Each of these approaches takes specific steps to ensure a topologically correct reconstruction so

that closed, nonintersecting surfaces with spherical topology (i.e. can be inflated to form a sphere) are formed. Manual interaction may be required in some cases to ensure accurate surface formation. A representation of the cerebral cortex has also been extracted using two level-set surfaces that have been coupled and simultaneously deformed [92].

2.1.3.2 Knowledge-based Segmentation

Knowledge can include anatomical knowledge, knowledge of image features, for example, the intensity or texture of objects in the image, or knowledge of the image formation process. This information can be included in several ways, such as with a full anatomical atlas that is warped to match the specific case under investigation, a shape description of a desired image object, or a set of rules defined by an expert and implemented in a rule-based algorithm.

2.1.3.2.1 Atlas-based Segmentation

An atlas is a composite head image formed from segmented, co-registered head images of several subjects. These have been used for segmentation by registration where a 3D mapping is determined between the atlas and the anatomy appearing in the image of interest and the labeling found in the atlas is used to identify the labeling applied to complete the segmentation [93], [94].

Atlases have been used to improve the selection of a priori probabilities for statistical pattern recognition [29], [33], [95]. In a further development, the required registration step has been incorporated into a statistical clustering framework that also performs simultaneous intensity nonuniformity correction where the expectation maximization method is used for parameter estimation and the model assumes that the pixel intensities

are drawn from a mixture of Gaussians [96]. Nonparametric classifiers, where no assumptions are made regarding the pixel intensity distributions, have also been used with atlases [91], [97], [98].

Atlas-based segmentation using statistical pattern recognition has been enhanced to include a model of the MR image formation process in order to reduce sensitivity to the image acquisition parameters. The integration of image acquisition into the process also facilitates the derivation of MR imaging sequences that are optimal for the purposes of segmentation [99].

A method that automatically matches cortical regions between an atlas and the image of interest and then performs a corresponding atlas-warping step to improve segmentation labeling is described in [100].

Although fully-automatic registration is desirable, semiautomatic registration is also used where manually-defined, landmark points constrain the deformation and improve segmentation accuracy especially in cortical regions where substantial inter-subject variability exists [101].

In a manner analogous to probabilistic atlases, fuzzy templates have been formed and used for segmentation of subcortical brain structures [102].

Atlases have also been used to provide anatomical knowledge without undergoing deformation [103].

2.1.3.2.2 Rule-based Segmentation

Automatic, rule-based guidance of unsupervised image segmentation has been explored in an attempt to improve the results from unsupervised segmentation methods and yet maintain an automated approach to the segmentation task. Image primitives are

usually derived from first-generation and second-generation algorithms and then interpreted using anatomical and image knowledge applied as a set of rules. The wide variety of methods that have been developed can usually be categorized by the types of image information that are extracted, by the types of knowledge that are introduced, and by the method used for implementation of that knowledge.

First-generation methods were often used in early, rule-based systems for image data extraction [104]-[108] and although these are relatively simple, low-level processing methods, the methods used for interpretation are not necessarily so. An oversegmented image and a rule-based merging algorithm is used in [105] to segment the lateral ventricles and left caudate nucleus in CT images of the brain. The process incorporated high-level domain knowledge, including an anatomical model.

A brain model describing the structures and features in each 2D slice of a 3D image for object recognition in CT images is described in [106]. This system is significant because of the use of a blackboard, a temporary database containing prior and current information, updated as the segmentation proceeds, and data fusion, combination of multiple data sources to improve decision confidence.

Second-generation methods have also been used to extract image primitives for rule-based systems [109]-[114]. An automatic system for segmentation and labeling of glioblastoma-multiforme tumours in MR images where gadolinium had been used as a contrast agent is described in [111] with further evaluation in [115]. Adaptive thresholds, fuzzy c-means clustering, and a rule-base built from a set of training images were key components of the multistage system. Of particular note is that all steps in the system were performed automatically.

Knowledge of anatomical shapes and relative object locations along with a hypothesize-and-verify approach driven by a genetic algorithm has been used to identify basal nuclei, ventricles, and other substructures in 2D, MR, brain images [110].

Possibilistic clustering, a generalization of fuzzy c-means clustering, and fuzzy logic have been used to segment the thalamus, putamen, and the head of the caudate nucleus from brain images of healthy volunteers [112]. The features used in the clustering process consist of a set of 3D wavelet coefficients extracted from the MR image at each pixel. The clustering is followed by a fuzzy logic step to refine the segmentation by incorporating linguistic descriptions, obtained from a human expert, describing object position and features.

A set of deformable mesh surfaces are simultaneously deformed, guided by a series of rules based on the dynamics of the evolving surfaces and medical knowledge of the shape and texture of the objects of interest, in [116]. The rules are implemented as constraints on the surface deformation.

2.1.3.2.3 Model-based Segmentation

Statistical model-based segmentation with nondeformable, shape models was explored as a guide to segmentation by region growing [117]; however, the popularity of statistical shape models has soared since the development of the active shape model (ASM) [118]. The ASM was inspired by active contours but with the added intention of limiting the extent of the model deformation to within reasonable limits for a given class of object, thus increasing the specificity of the model. Since objects of a given class display shape variability, a point distribution model (PDM) is formed using points attached to specific landmarks on a particular object. In the PDM, each landmark point has a statistical

distribution determined from a set of N co-registered, example objects, ideally describing the full range of shape variability. Assuming 2D coordinates and n landmark points in the PDM, the ASM can be viewed as a point cloud of N points in a $2n$-dimensional space. A mean contour is determined and the deviation from the mean is computed for each of the N contours and a $2n \times 2n$ covariance matrix is formed. Principal components analysis is then used to identify the modes of greatest variability in this high-dimensional space.

The ASM, thus trained, is then used to identify objects of the same class within other images. The process is initialized by identifying an orientation and scale for the mean contour and then permitting it to deform within the image in a manner analogous to that of active contours. In this case, the movement of each landmark point is to the nearest intensity gradient in a direction normal to the contour. Deformation is restricted to within the bounds of the statistical model, usually held to within three standard deviations from the mean.

Related, shape-based approaches to image segmentation have also been developed [119]-[123] and other shape representations, spherical harmonics [124], wavelet [125], shape-variant Hough transform [126], and medial (skeleton) representation [127], have also been used. A level set shape representation that does not require explicitly defined PDMs, and that also permits surface topology modification as the model deforms, has been introduced [128], [129]. This has been further extended to permit simultaneous, coupled deformation of multiple shape models [130]. Other deformable, statistical shape models called strings and necklaces have been used for segmentation of spinal images [131].

An active appearance model (AAM) is a further development of ASMs where shape plus intensity of an object, referred to as an image patch, and corresponding variability are integrated into a statistical model [132]. A hybrid combination of ASMs and AAMs was used to develop an automated segmentation method for MR, cardiac images [133]. A level set implementation of a shape-plus-intensity model in 3D is described in [134].

Artificial organisms have also been applied to segmentation of MR images [135]. These represent further developments of deformable models. They include a body capable of taking the shape of the desired image object and having distributed sensors and rudimentary brains with centres for perception, cognition, and motor control. They are capable of voluntary movement and alteration of body shape based on sensory input from the image and anatomical knowledge. As such, the artificial organism released into the image environment will seek out and conform to the object of interest.

2.2 Segmentation Software

Medical image segmentation software is available commercially from a number of sources, typically as a subset of a larger software package intended for general, interactive, image processing. Also, over the past few years, the amount of free software that has become available has grown rapidly. Supported by government, academic institutions, or individuals, this software often includes source-level program code (high–level computer programming language implementation), referred to as open source. A sampling of the available software is given in the following sections. Additionally, the Neuro Image Analysis group at the University of North Carolina at Chapel Hill maintains a webpage with links to a number of related sites where free software can be downloaded [136].

2.2.1 BIC Software Toolbox

The McConnell Brain Imaging Centre (BIC) of the Montreal Neurological Institute (MNI) at McGill University freely offers a variety of software for medical image analysis [137]. Included are tools for automatic registration (ANIMAL), segmentation (INSECT), intensity nonuniformity correction (N3), sulcus extraction and labeling (SEAL), and cortex extraction. Tools for PET and fMRI analysis are also available as are a number of display and manual, image-editing functions. A description of some of the software modules and a technique for application to the analysis of large image databases is described in [138]

Source code is available. All image data are stored in Medical Image NetCDF (MINC) file format. An interface to Matlab® (EMMA) is also available.

2.2.2 SPM

Statistical Parametric Mapping (SPM) developed at the Wellcome Department of Cognitive Neurology, University College of London, England, is freely available software intended for analysis of brain imaging data sequences [139]. Supported modalities include fMRI, PET, and SPECT with analysis of EEG and Magnetoencephalogram (MEG) expected in future releases. Segmentation in SPM is performed using unsupervised, statistical clustering.

2.2.3 FSL

The FMRIB Software Library (FSL) was developed by the Oxford Center for Functional Magnetic Resonance Imaging of the Brain (FMRIB) at the University of Oxford, England. The software is a library of statistical and image analysis tools for

fMRI, structural MRI, and Diffusion Tensor Imaging (DTI) data. Segmentation is performed by application of an automated, brain extraction tool (BET) based on a deformable model. This is followed by automatic segmentation of gray matter, white matter, and cerebrospinal fluid using the hidden Markov random field and expectation–maximization, statistical clustering approach of [32]. The FSL software is freely available for noncommercial use [140].

2.2.4 MEDx

MEDx is a commercial software package developed by Sensor Systems Medical Products Division [141] located in Sterling, Virginia, USA. It is intended for multimodality, multidimensional, image analysis. Both SPM and FSL are used in MEDx.

2.2.5 EIKONA3D

Eikona3D is a commercial software package developed by Alpha Tec, Ltd. [142], located in Thessaloniki, Greece. The software contains a core set of image processing and visualization tools. Interactive image segmentation can be performed by threshold and by region-based methods such as region growing and region split-merge. Basic edge detection and edge-following features are also available.

2.2.6 FreeSurfer

FreeSurfer [85] is a software package developed by CorTechs Labs [143] and the Athinoula A. Martinos Center for Biomedical Imaging at the Massachusetts General Hospital, an affiliate of Harvard University [144]. The software employs automatic and manual segmentation methods for semiautomatic reconstruction of the cerebral cortex

from structural, MR images and also allows overlay of fMRI and EEG data onto the reconstructed surface.

2.2.7 Insight Segmentation and Registration Toolkit

The insight segmentation and registration toolkit (ITK) [145] and its companion the visualization toolkit (VTK) [146] have been developed by the United States National Library of Medicine in support of the Visible Human Project. Development began in 1999 and is ongoing. The toolkits are open-source, freely available, software modules written in the C++ computer language and supported by Kitware [147], a professional software development corporation, and also by qualified volunteers. The software modules are intended for use in image segmentation and registration, computer-software applications but are not full-fledged applications, themselves.

The segmentation modules include a variety of low-level methods such as thresholds and region growing as well as higher-level segmentation using the watershed algorithm and deformable models including those based on the level set approach.

A stand-alone, segmentation software application called SNAP, for Snake Automated Partitioning, was developed using level set algorithms in ITK and is freely distributed as open-source software [148].

2.2.8 Analyze

Analyze is a commercial, image analysis, software package developed by the Biomedical Imaging Resource at the Mayo Foundation [149] based in Rochester, Minnesota, USA. It may be "the most comprehensive, robust and productive software package available for 3D biomedical image visualization and analysis" as their website

suggests. Certainly it appears to have the longest history, beginning in the early 1970's, and has impressive capabilities for volume and surface visualization, including the ability to generate cine fly-through image sequences for virtual endoscopy. These visualization steps would typically be performed after extraction of objects from an image by segmentation.

The software permits automatic segmentation using modules from ITK, including segmentation based on level sets, fuzzy connectedness, and the watershed algorithm. Interactive, manual segmentation and region growing can also be performed and a number of lower-level filtering functions are available.

2.2.9 3D Slicer

The 3D Slicer software [150] is developed by the MIT Artificial Intelligence Lab and the Surgical Planning Lab at Brigham and Women's Hospital, an affiliate of Harvard Medical School. It is open-source, freely-available software based on VTK. It contains functions for 3D image registration, as well as formation, visualization, and quantification of 3D surfaces and volumes. Segmentation is performed via manual and semiautomatic means, including thresholds, binary morphology (erosion and dilation), and manual and semiautomatic line drawing. A module that performs automatic statistical clustering using the expectation-maximization algorithm is also included.

2.3 References

[1] J.C. Bezdek, L.O. Hall, L.P. Clarke, "Review of MR image segmentation techniques using pattern recognition," *Medical Physics,* Vol. 20, No. 4, 1993, pp. 1033-1048.

[2] L.P. Clarke, R.P. Velthuizen, M.A. Camacho, J.J. Heine, M. Vaidyanathan, L.O. Hall, R.W. Thatcher, M.L. Silbiger, "MRI segmentation: Methods and applications," *Magnetic Resonance Imaging*, Vol. 13, No. 3, 1995, pp. 343-368.

[3] S.D. Olabarriaga, A.W.M. Smeulders, "Interaction in the segmentation of medical images: A survey," *Medical Image Analysis*, Vol. 5, 2001, pp. 127-142.

[4] D.L. Pham, C. Xu, J.L. Prince, "Current methods in medical image segmentation," *Annual Review of Biomedical Engineering*, Vol. 2, 2000, pp. 315-337.

[5] J.S. Duncan, N. Ayache, "Medical image analysis: Progress over two decades and the challenges ahead," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 1, 2000, pp. 85-106.

[6] E.M. Haacke, Z-P. Liang, "Challenges of imaging structure and function with MRI," *IEEE Engineering in Medicine and Biology*, September/October, 2000, pp. 55-62.

[7] A.K. Jain, R.P.W. Duin, J. Mao, "Statistical pattern recognition: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 1, 2000, pp. 4-37.

[8] J.C. Bezdek, L.O. Hall, M.C. Clark, D.B. Goldgof, L.P. Clarke, "Medical image analysis with fuzzy models," *Statistical Methods in Medical Research*, Vol. 6, 1997, pp. 191-214.

[9] T. McInerney, D. Terzopoulos, "Deformable models in medical image analysis: A survey," *Medical Image Analysis*, Vol. 1, No. 2, 1996, pp. 91-108.

[10] A.K. Jain, Y. Zhong, M.-P. Dubuisson-Jolly, "Deformable template models: A review," *Signal Processing*, Vol. 71, 1998, pp. 109-129.

[11] J.S. Suri, S. Singh, L. Reden, "Computer vision and pattern recognition techniques for 2-D and 3-D MR cerebral cortical segmentation (Part 1): A state-of-the-art review," *Pattern Analysis and Applications*, Vol. 5, 2002, pp.46-76.

[12] J.S. Suri, S. Singh, L. Reden, "Fusion of region and boundary/surface-based computer vision and pattern recognition techniques for 2-D and 3-D MR cerebral cortical segmentation (Part 2): A state-of-the-art review," *Pattern Analysis and Applications*, Vol. 5, 2002, pp.77-98.

[13] I. Pitas, "Digital Image Processing Algorithms," Prentice-Hall, 1993.

[14] N. Nikolaidis, I. Pitas, "3-D Image Processing Algorithms," John Wiley & Sons, Inc., 2001.

[15] W.K. Pratt, "Digital Image Processing," John Wiley & Sons, Inc., 1991.

[16] A.R. Weeks. "Fundamentals of Electronic Image Processing," SPIE Optical Engineering Press, IEEE Press (copublishers), 1996.

[17] D. Kim, J. Park, "Computerized quantification of carotid artery stenosis using MRA axial images," *Magnetic Resonance Imaging*, Vol. 22, 2004, pp. 353-359.

[18] S.G. Armato III, W.F. Sensakovic, "Automated lung segmentation for thoracic CT: Impact on computer aided diagnosis," *Academic Radiology*, Vol. 11, No. 9, 2004, pp. 1011-1021.

[19] S.-Y. Wan, W.E. Higgins, "Symmetric region growing," *IEEE Transactions on Image Processing*, Vol. 12, No. 9, 2003, pp. 1007-1015.

[20] S.A. Hojjatoleslami, J. Kittler, "Region growing: A new approach," *IEEE Transactions on Image Processing*, Vol. 7, No. 7, 1998, pp. 1079-1084.

[21] S.A. Hojjatoleslami, F. Kruggel, "Segmentation of large brain lesions," *IEEE Transactions on Medical Imaging*, Vol. 20, No. 7, 2001, pp. 666-669.

[22] M. Lineberry, "Image segmentation by edge tracing," in Proc. of SPIE, The International Society for Optical Engineering, Vol. 359, *Applications of Digital Image Processing 4*, San Diego, Calif., USA, 1982, pp. 361-368.

[23] A. Martelli, "An application of heuristic search methods to edge and contour detection," *Communications of the ACM*, Vol. 19, No. 2, 1976, pp. 73-83.

[24] J.D. Cappelletti, A. Rosenfeld, "Three-dimensional boundary following," *Computer Vision, Graphics, and Image Processing*, Vol. 48, 1989, pp. 80-92.

[25] X. Qu, X. Li, "A 3D surface tracking algorithm," *Computer Vision and Image Understanding*, Vol. 64, No. 1, 1996, pp. 147-156.

[26] P. Anbeek, K.L. Vincken, G.S. van Bochove, M.J.P. van Osch, J. van der Grond, "Probabilistic segmentation of brain tissue in MR imaging," *NeuroImage*, Vol. 27, 2005, pp. 795-804.

[27] W.M. Wells III, W.E.L. Grimson, R. Kikinis, F.A. Jolesz, "Adaptive segmentation of MRI data," *IEEE Transactions on Medical Imaging*, Vol. 15, No. 4, 1996, pp. 429-442.

[28] K. Held, E. Rota Kops, B.J. Krause, W.M. Wells III, R. Kikinis, H.-W. Müller-Gärtner, "Markov random field segmentation of brain MR images," *IEEE Transactions on Medical Imaging*, Vol. 16, No. 6, 1997, pp. 878-886.

[29] K.Van Leemput, F. Maes, D. Vandermeulen, P. Suetens, "Automated model-based tissue classification of MR images of the brain," *IEEE Transactions on Medical Imaging*, Vol. 18, No. 10, 1999, pp. 897-908.

[30] R. Chellappa, A. Jain, eds., "Markov Random Fields Theory and Application," Academic Press, 1993.

[31] K. Van Leemput, F. Maes, D. Vandermeulen, P. Suetens, "A unifying framework for partial volume segmentation of brain MR images," *IEEE Transactions on Medical Imaging*, Vol. 22, No. 1, 2003, pp. 105-119.

[32] Y. Zhang, M. Brady, S. Smith, "Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm," *IEEE Transactions on Medical Imaging*, Vol. 20, No. 1, 2001, pp. 45-57.

[33] J.L. Marroquin, B.C. Vemuri, S. Botello, F. Calderon, A. Fernandez-Bouzas, "An accurate and efficient Bayesian method for automatic segmentation of brain MRI," *IEEE Transactions on Medical Imaging*, Vol. 21, No. 8, 2002, pp. 934-945.

[34] S.-K. Ng, G.J. McLachlan, "Speeding up the EM algorithm for mixture model-based segmentation of magnetic resonance images," *Pattern Recognition*, Vol. 37, 2004, pp. 1573-1589.

[35] J.L. Marroquin, E.A. Santana, S. Botello, "Hidden Markov measure field models for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 11, 2003, pp. 1380-1387.

[36] J. Tohka, A. Zijdenbos, A.C. Evans, "Fast and robust parameter estimation for statistical partial volume models in brain MRI," *NeuroImage*, Vol. 23, 2004, pp. 84-97.

[37] M.Á.González Ballester, A.P. Zisserman, M. Brady, "Estimation of the partial volume effect in MRI," *Medical Image Analysis*, Vol. 6, 2002, pp. 389-405.

[38] D.L. Pham, J.L. Prince, "Adaptive fuzzy segmentation of magnetic resonance images," *IEEE Transactions on Medical Imaging*, Vol. 18, No. 9, 1999, pp. 737-752.

[39] M.N. Ahmed, S.M. Yamany, N. Mohamed, A.A. Farag, T. Moriarty, "A modified fuzzy c-means algorithm for bias field estimation and segmentation of MRI data," *IEEE Transactions on Medical Imaging*, Vol. 21, No. 3, 2002, pp. 193-199.

[40] D.-Q. Zhang, S.-C. Chen, "A novel kernalized fuzzy c-means algorithm with application in medical image segmentation," *Artificial Intelligence in Medicine*, Vol. 32, 2004, pp. 37-50.

[41] A.W.-C. Liew, H. Yan, "An adaptive spatial fuzzy clustering algorithm for 3-D MR image segmentation," *IEEE Transactions on Medical Imaging*, Vol. 22, No. 9, 2003, pp. 1063-1075.

[42] A. Rosenfeld, "On connectivity properties of grayscale pictures," *Pattern Recognition,* Vol. 16, No. 1, 1983, pp. 47-50.

[43] A. Rosenfeld, "The fuzzy geometry of image subsets," *Pattern Recognition Letters*, Vol. 2, 1984, pp. 311-317.

[44] J.K. Udupa, S. Samarasekera, "Fuzzy connectedness and object definition: Theory, algorithms, and applications in image segmentation," *Graphical Models and Image Processing*, Vol. 58, No. 3, 1996, pp. 246-261.

[45] J.K. Udupa, P.K. Saha, "Fuzzy connectedness and image segmentation," *Proceedings of the IEEE,* Vol. 91, No. 10, 2003, pp. 1649-1669.

[46] N. Hata, Y. Muragaki, T. Inomata, T. Maruyama, H. Iseki, T. Hori, T. Dohi, "Intraoperative tumor segmentation and volume measurement in MRI-guided glioma surgery for tumor resection rate control," *Academic Radiology,* Vol. 12, No. 1, 2005, pp. 116-122.

[47] M. Kass, A. Witkin, D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, Vol. 1, No. 4, 1988, pp. 321-331.

[48] T. McInerney, D. Terzopoulos, "Topology adaptive deformable surfaces for medical image volume segmentation," *IEEE Transactions on Medical Imaging*, Vol. 18, No. 10, 1999, pp. 840-850.

[49] R.G.N. Meegama, J.C. Rajapakse, "NURBS snakes," *Image Vision and Computing*, Vol. 21, 2003, pp. 551-562.

[50] M. Wei, Y. Zhou, M. Wan, "A fast snake model based on non-linear diffusion for medical image segmentation," *Computerized Medical Imaging and Graphics*, Vol. 28, 2004, pp. 109-117.

[51] J. Wang, X. Li, "Guiding ziplock snakes with a priori information," *IEEE Transactions on Image Processing*, Vol. 12, No. 2, 2003, pp. 176-185.

[52] R. Malladi, J.A. Sethian, B.C. Vemuri, "Shape modeling with front propagation: A level set approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 2, 1995, pp. 158-175.

[53] V. Caselles, R. Kimmel, G. Sapiro, "Geodesic active contours," *International Journal of Computer Vision*, Vol. 22, No. 1, 1997, pp. 61-79.

[54] V. Caselles, R. Kimmel, G. Sapiro, C. Sbert "Minimal surfaces based object segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 4, 1997, pp. 394-398.

[55] J.A. Sethian, "Level set methods and fast marching methods," Cambridge University Press, 1999.

[56] M. Xu, P.M. Thompson, A.W. Toga, "An adaptive level set segmentation on a triangulated mesh," *IEEE Transactions on Medical Imaging*, Vol. 23, No. 2, 2004, pp. 191-201.

[57] D. Mumford, J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Communications on Pure and Applied Mathematics*, Vol. 42, No. 5, 1989, pp. 577-685.

[58] T.F. Chan, L.A. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, Vol. 10, No. 2, 2001, pp. 266-277.

[59] A. Tsai, A. Yezzi, Jr., A.S. Willsky, "Curve evolution implementation of the Mumford-Shah functional for image segmentation, denoising, interpolation, and magnification," *IEEE Transactions on Image Processing*, Vol. 10, No. 8, 2001, pp. 1169-1186.

[60] B. Appleton, H. Talbot, "Globally minimal surfaces by continuous maximal flows," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, No. 1, 2006, pp. 106-118.

[61] L. Vincent, P. Soille, Watersheds in digital spaces: An efficient algorithm based on immersion simulations, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 6, 1991, pp. 583-598.

[62] J. Sijbers, P. Scheunders, M. Verhoye, A. Van der Linden, D. van Dyck, E. Raman, "Watershed-based segmentation of 3D MR data for volume quantization," *Magnetic Resonance Imaging*, Vol. 15, No. 6, 1997, pp. 679-688.

[63] V. Barrios, J. Torres, G. Montilla, L. Hernandez, N. Rangel, A. Reigosa, "Cellular edge detection using a trained neural network explorer," in Proc. 16th Annual

International Conference IEEE Engineering in Medicine and Biology Society, Vol. 2, 1994, pp. 1075-1076.

[64] R. Eckhorn, "Neural mechanisms of scene segmentation: Recordings from the visual cortex suggest basic circuits for linking field models," *IEEE Transactions on Neural Networks*, Vol. 10, No. 3, 1999, pp. 464-479.

[65] J.L. Johnson, M.L. Padgett, "PCNN models and applications," *IEEE Transactions on Neural Networks*, Vol. 10, No. 3, 1999, pp. 480-498.

[66] N. Shareef, D.L. Wang, R. Yagel, "Segmentation of medical images using LEGION," *IEEE Transactions on Medical Imaging*, Vol. 18, No. 1, 1999, pp. 74-91.

[67] A. Wismuller, F. Vietze, J. Behrends, A. Meyer-Baese, M. Reiser, H. Ritter, "Fully automated biomedical image segmentation by self-organized model adaptation," *Neural Networks*, Vol. 17, 2004, pp. 1327-1344.

[68] U. Raff, F.D. Newman, "Automated lesion detection and lesion quantitation in MR images using autoassociative memory," *Medical Physics*, Vol. 19, No. 1, 1992, pp. 71-77.

[69] V. Boskovitz, H. Guterman, "An adaptive neuro-fuzzy system for automatic image segmentation and edge detection," *IEEE Transactions on Fuzzy Systems*, Vol. 10, No. 2, 2002, pp. 247-262.

[70] H. Soltanian-Zadeh, J.P. Windham, "A multiresolution approach for contour extraction from brain images," *Medical Physics*, Vol. 24, No. 12, 1997, pp. 1844-1853.

[71] S.V. Raman, S. Sarkar, K.L. Boyer, "Tissue boundary refinement in magnetic resonance images using contour-based scale space matching," *IEEE Transactions on Medical Imaging*, Vol. 10, No. 2, 1991, pp. 109-121.

[72] K.L. Vincken, A.S.E. Koster, M.A. Viergever, "Probabilistic multiscale image segmentation," *IEEE Transactions Pattern Analysis and Machine Intelligence*, Vol. 19, No. 2, 1997, pp. 109-120.

[73] W.J. Neissen, K.L Vinken, J. Weickert, B.M. Ter Haar Romeny, M.A. Viergever, "Multiscale segmentation of three-dimensional MR brain images," *International Journal of Computer Vision*, Vol. 31, 1999, pp. 185-202.

[74] A.X. Falcão, J.K. Udupa, F.K. Miyazawa, "An ultra-fast user-steered image segmentation paradigm: Live wire on the fly," *IEEE Transactions on Medical Imaging*, Vol. 19, No. 1, 2000, pp. 55-62.

[75] A.X. Falcão, J.K. Udupa, S. Samarasekera, S. Sharma, "User-steered image segmentation paradigms: Live wire and live lane," *Graphical Models and Image Processing*, Vol. 60, 1998, pp. 233-260.

[76] A.X. Falcão, J.K. Udupa, "A 3D generalization of user-steered live-wire segmentation," *Medical Image Analysis*, Vol. 4, 2000, pp. 389-402.

[77] D.H. Chung, G. Sapiro, "Segmenting skin lesions with partial-differential-equations-based image processing algorithms," *IEEE Transactions on Medical Imaging*, Vol. 19, No. 7, 2000, pp. 763-767.

[78] M.M.J Letteboer, O.F. Olsen, E.B. Dam P.W.A. Willems, M.A. Viergever, W.J. Niessen, "Segmentation of tumors in magnetic resonance brain images using an interactive multiscale watershed algorithm," *Academic Radiology*, Vol. 11, 2004, pp. 1125-1138.

[79] I. Middleton, R.I. Damper, "Segmentation of magnetic resonance images using a combination of neural networks and active contour models," *Medical Engineering and Physics*, Vol. 26, 2004, pp. 71-86.

[80] L. Amini, H. Soltanian-Zadeh, C. Lucas, M. Gity, "Automatic segmentation of thalamus from brain MRI integrating fuzzy clustering and dynamic contours," *IEEE Transactions on Biomedical Engineering*, Vol. 51, No. 5, 2004, pp. 800-811.

[81] T. Kapur, W.E.L. Grimson, W.M. Wells III, R. Kikinis, "Segmentation of brain tissue from magnetic resonance images," *Medical Image Analysis*, Vol. 1, No. 2, 1996, pp. 109-127.

[82] E.A. Ashton, M.J. Berg, K.J. Parker, J. Weisberg, C.W. Chen, L. Ketonen, "Segmentation and feature extraction techniques with applications to MRI head studies," *Magnetic Resonance in Medicine*, Vol. 33, 1995, pp. 670-677.

[83] A. Chakraborty, L.H. Staib, J.S. Duncan, "Deformable boundary finding in medical images by integrating gradient and region information," *IEEE Transactions on Medical Imaging*, Vol. 15, No. 6, 1996, pp. 859-870.

[84] L.S. Hibbard, "Region segmentation using information divergence measures," *Medical Image Analysis*, Vol. 8, 2004, pp. 233-244.

[85] http://surfer.nmr.mgh.harvard.edu

[86] A.M. Dale, B. Fischl, M.I. Sereno, "Cortical surface-based analysis I: Segmentation and surface reconstruction," *NeuroImage*, Vol. 9, 1999, pp. 179-194.

[87] B. Fischl, M.I. Sereno, A.M. Dale, "Cortical surface-based analysis II:. Inflation, flattening, and a surface-based coordinate system," *NeuroImage*, Vol. 9, 1999, pp. 195-207.

[88] B. Fischl, A. Liu, A.M. Dale, "Automated manifold surgery: Constructing geometrically accurate and topologically correct models of the human cerebral cortex," *IEEE Transactions on Medical Imaging*, Vol. 20, No. 1, 2001, pp. 70-80.

[89] F. Ségonne, A.M. Dale, E. Busa, M. Glessner, D. Salat, H.K. Hahn, B. Fischl, "A hybrid approach to the skull stripping problem in MRI," *NeuroImage*, Vol. 22, 2004, pp. 1060-1075.

[90] D. Tosun, M.E. Rettmann, X. Han, X. Tao, C. Xu, S.M. Resnick, D.L. Pham, J.L. Prince, "Cortical surface segmentation and mapping," *NeuroImage*, Vol. 23, 2004, pp. S108-S118.

[91] J.S. Kim, V. Singh, J.K. Lee, J. Lerch, Y. Ad-Dab'bagh, D. MacDonald, J.M. Lee, S.I. Kim, A.C. Evans, "Automated 3-D extraction and evaluation of the inner and outer cortical surfaces using a Laplacian map and partial volume effect classification," *NeuroImage*, Vol. 27, 2005, pp. 210-221.

[92] X. Zeng, L.H. Staib, R.T. Schultz, J.S. Duncan, "Segmentation and measurement of the cortex from 3-D MR images using coupled-surfaces propagation," *IEEE Transactions on Medical Imaging*, Vol. 18, No. 10, 1999, pp. 927-937.

[93] D.L. Collins, C.J. Holmes, T.M. Peters, A.C. Evans, "Automatic 3-D model-based neuroanatomical segmentation," *Human Brain Mapping*, Vol. 3, No. 3, 1995, pp. 190-208.

[94] B.M. Dawant, S.L. Hartmann, J.-P. Thirion, F. Maes, D. Vandermeulen, P. Demaerel, "Automatic 3-D segmentation of internal structures of the head in MR images using a combination of similarity and free-form transformations: Part I, Methodology and

validation on normal subjects," *IEEE Transactions on Medical Imaging*, Vol. 18, No. 10, 1999, pp. 909-916.

[95] J. Ashburner, K. Friston, Multimodal image coregistrations and partitioning – A unified framework, *NeuroImage*, Vol. 6, No. 3, 1997, pp. 209-217.

[96] J. Ashburner, K.J. Friston, "Unified segmentation," *NeuroImage*, Vol. 26, 2005, pp. 839-851.

[97] C.A. Cocosco, A.P. Zijdenbos, A.C. Evans, "A fully automatic and robust brain MRI tissue classification method," *Medical Image Analysis*, Vol. 7, 2003, pp. 513-527.

[98] J.R. Jiménez-Alaniz, V. Medina-Bañuelos, O. Yáñez-Suárez, "Data-driven brain MRI segmentation supported on edge confidence and a priori tissue information," *IEEE Transactions on Medical Imaging*, Vol. 25, No. 1, 2006, pp. 74-83.

[99] B. Fischl, D.H. Salat, A.J.W. van der Kouwe, N. Makris, F. Ségonne, B.T. Quinn, A.M. Dale, "Source-independent segmentation of magnetic resonance images," *NeuroImage*, Vol. 23, 2004, pp. S69-S84.

[100] A. Klein, J. Hirsch, "Mindboggle: A scatterbrained approach to automate brain labeling," *NeuroImage*, Vol. 24, 2005, pp. 261-280.

[101] L.A. Dade, F.Q. Gao, N. Kovacevic, P. Roy, C. Rockel, C.M. O'Toole, N.J. Lobaugh, A. Feinstein, B. Levine, S.E. Black, "Semiautomatic brain region extraction: A method of parcellating brain regions from structural magnetic resonance images," *NeuroImage*, Vol. 22, 2004, pp. 1492-1502.

[102] J. Zhou, J.C. Rajapakse, "Segmentation of subcortical brain structures using fuzzy templates," *NeuroImage*, Vol. 28, 2005, pp. 915-924.

[103] A.J. Worth, N. Makris, M.R. Patti, J.M. Goodman, E.A. Hoge, V.S. Caviness, Jr., D.N. Kennedy, "Precise segmentation of the lateral ventricles and caudate nucleus in MR brain images using anatomically driven histograms," *IEEE Transactions on Medical Imaging*, Vol. 17, No. 2, 1998, pp. 303-310.

[104] H.-H. Ehricke, "Automated 3D MR image analysis with a rule-based segmentation system," in: CAR '91, Computer Assisted Radiology, Berlin, Germany: Springer Verlag, 1991, pp. 543-548.

[105] K. Natarajan, M.G. Cawley, J.A. Newell, "A knowledge-based system paradigm for automatic interpretation of CT scans," *Medical Informatics*, Vol. 16, No. 2, 1991, pp. 167-181.

[106] H. Li, R. Deklerck, B. De Cuyper, A. Hermanus, E. Nyssen, J. Cornelis, "Object recognition in brain CT-scans: Knowledge-based fusion of data from multiple feature extractors," *IEEE Transactions on Medical Imaging*, Vol. 14, No. 2, 1995, pp. 212-229.

[107] H. Suzuki, J. Toriwaki, "Automatic segmentation of head MRI images by knowledge guided thresholding," *Computerized Medical Imaging and Graphics*, Vol. 15, No. 4, 1991, pp. 233-240.

[108] S.P. Raya, "Low-level segmentation of 3-D magnetic resonance brain images – A rule-based system," *IEEE Transactions on Medical Imaging*, Vol. 9, No. 3, 1990, pp. 327-337.

[109] B. Solaiman, R. Debon, F. Pipelier, J.-M. Cauvin, C. Roux, "Information fusion: Application to data and model fusion for ultrasound image segmentation," *IEEE Transactions on Biomedical Engineering*, Vol. 46, No. 10, 1999, pp. 1171-1175.

[110] M. Sonka, S.K. Tadikonda, S.M. Collins, "Knowledge-based interpretation of MR brain images," *IEEE Transactions on Medical Imaging*, Vol. 15, No. 4, 1996, pp. 443-452.

[111] M.C. Clark, L.O. Hall, D.B. Goldgof, R. Velthuizen, F.R. Murtagh, M.S. Silbiger, "Automatic tumor segmentation using knowledge-based techniques," *IEEE Transactions on Medical Imaging*, Vol. 17, No. 2, 1998, pp. 187-201.

[112] V. Barra, J.-Y. Boire, "Automatic segmentation of subcortical brain structures in MR images using information fusion," *IEEE Transactions on Medical Imaging*, Vol. 20, No. 7, 2001, pp. 549-558.

[113] C.W. Chen, J. Luo, K.J. Parker, "Image segmentation via adaptive K-mean clustering and knowledge-based morphological operations with biomedical applications," *IEEE Transactions on Image Processing*, Vol. 7, No. 12, 1998, pp. 1673-1683.

[114] M.-E. Algorri, F. Flores-Mangas, "Classification of anatomical structures in MR brain images using fuzzy parameters," *IEEE Transactions on Biomedical Engineering*, Vol. 51, No. 9, 2004, pp. 1599-1608.

[115] G.P. Mazzara, R.P. Velthuizen, J.L. Pearlman, H.M. Greenberg, H. Wagner, "Brain tumor target volume determination for radiation treatment planning through automated MRI segmentation," *International Journal of Radiation Oncology Biology Physics*, Vol. 59, No. 1, 2004, pp. 300-312.

[116] A. Pitiot, H. Delingette, P.M. Thompson, N. Ayache, "Expert knowledge-guided segmentation system for brain MRI," *NeuroImage*, Vol. 23, 2004, pp. S85-S96.

[117] L.K. Arata, A.P. Dhawan, J.P. Broderick, M.F. Gaskil-Shipley, A.V. Levy, N.D. Volkow, "Three-dimensional anatomical model-based segmentation of MR brain images

through principal axes registration," *IEEE Transactions on Biomedical Engineering*, Vol. 42, No. 11, 1995, pp. 1069-1078.

[118] T.F. Cootes, C.J. Taylor, D.H. Cooper, J. Graham, "Active shape models – Their training and application," *Computer Vision and Image Understanding*, Vol. 61, No. 1, 1995, pp. 38-59.

[119] N. Duta, M. Sonka, "Segmentation and interpretation of MR brain images: An improved active shape model," *IEEE Transactions on Medical Imaging*, Vol. 17, No. 6, 1998, pp. 1049-1062.

[120] A. Pitiot, A.W. Toga, P.M. Thompson, "Adaptive elastic segmentation of brain MRI via shape-model-guided evolutionary programming," *IEEE Transactions on Medical Imaging*, Vol. 21, No. 8, 2002, pp. 910-923.

[121] B. van Ginneken, A.F. Frangi, J.J. Staal, B. M. ter Haar Romeny, M.A. Viergever, "Active shape model segmentation with optimal features," *IEEE Transactions on Medical Imaging*, Vol. 21, No. 8, 2002, pp. 924-933.

[122] D. Shen, E.H. Herskovits, C. Davatzikos, "An adaptive-focus statistical shape model for segmentation and shape modeling of 3-D brain structures," *IEEE Transactions on Medical Imaging*, Vol. 20, No. 4, 2001, pp. 257-270.

[123] B. Olstad, A.H. Torp, "Encoding of a priori information in active contour models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18, No. 9, 1996, pp. 863-872.

[124] A. Kelemen, G. Székely, G. Gerig, "Elastic model-based segmentation of 3-D neuroradiological data sets," *IEEE Transactions on Medical Imaging*, Vol. 18, No. 10, 1999, pp. 828-839.

[125] C. Davatzikos, X. Tao, D. Shen, "Hierarchical active shape models, using the wavelet transform," *IEEE Transactions on Medical Imaging*, Vol. 22, No. 3, 2003, pp. 414-423.

[126] M. Brejl, M. Sonka, "Object localization and border detection criteria design in edge-based image segmentation: Automated learning from examples," *IEEE Transactions on Medical Imaging*, Vol. 19, No. 10, 2000, pp. 973-985.

[127] S. Joshi, S. Pizer, P.T. Fletcher, P. Yushkevich, A. Thall, J.S. Marron, "Multiscale deformable model segmentation and statistical shape analysis using medial descriptions," *IEEE Transactions on Medical Imaging*, Vol. 21, No. 5, 2002, pp. 538-550.

[128] M.E. Leventon, W.E.L. Grimson, O. Faugeras, "Statistical shape influence in geodesic active contours," in: IEEE Conference on Computer Vision and Pattern Recognition, Vol. 1, 2000, pp. 316-323.

[129] A. Tsai, A. Yezzi, Jr., W. Wells, C. Tempany, D. Tucker, A. Fan, W.E. Grimson, A. Willsky, "A shape-based approach to the segmentation of medical imagery using level sets," *IEEE Transactions on Medical Imaging*, Vol. 22, No. 2, 2003, pp. 137-154.

[130] A. Tsai, W. Wells, C. Tempany, E. Grimson, A. Willsky, "Mutual information in coupled multi-shape model for medical image segmentation," *Medical Image Analysis*, Vol. 8, 2004, pp. 429-445.

[131] S. Ghebreab, A.W.M. Smeulders, "Combining strings and necklaces for interactive three-dimensional segmentation of spinal images using an integral deformable spine model," *IEEE Transactions on Biomedical Engineering*, Vol. 51, No. 10, 2004, pp. 1821-1829.

[132] T.F. Cootes, G.J. Edwards, C.J. Taylor, "Active appearance models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 6, 2001, pp. 681-685.

[133] S.C. Mitchell, B.P.F. Lelieveldt, R.J. van der Geest, H.G. Bosch, J.H.C. Reiber, M. Sonka, "Multistage hybrid active appearance model matching: Segmentation of left and right ventricles in cardiac MR images," *IEEE Transactions on Medical Imaging*, Vol. 20, No. 5, 2001, pp. 415-423.

[134] J. Yang, J.S. Duncan, "3D image segmentation of deformable objects with joint shape-intensity prior models using level sets," *Medical Image Analysis*, Vol. 8, 2004, pp. 285-294.

[135] T. McInerney, G. Hamarneh, M. Shenton, D. Terzopoulos, "Deformable organisms for automatic medical image analysis," *Medical Image Analysis*, Vol. 6, 2002, pp. 251-266.

[136] http://www.ia.unc.edu/dev/download/

[137] http://www.bic.mni.mcgill.ca/software/

[138] A.C. Evans, "Large-scale morphometric analysis of neuroanatomy and neuropathology," *Anatomy and Embryology*, Vol. 210, 2005, pp. 439-446.

[139] http://www.fil.ion.ucl.ac.uk/spm/

[140] http://www.fmrib.ox.ac.uk/fsl/

[141] http://medx.sensor.com/products/medx/overview.html

[142] http://www.alphatecltd.com/eikona3d/imageprocessing3d.html

[143] http://www.cortechs.net/

[144] http://www.nmr.mgh.harvard.edu/martinos/flashHome.php

[145] http://www.itk.org

[146] http://public.kitware.com/VTK/

[147] http://www.kitware.com

[148] http://www.itksnap.org/

[149] http://www.mayo.edu/bir/Software/Analyze/Analyze1.html

[150] http://www.slicer.org/

Chapter 3

Dynamic Edge Tracing for 2D Image Segmentation[1]

3.1 Introduction

Magnetic Resonance (MR) images are excellent sources of patient-specific, anatomical information. Automatic segmentation of these images into component tissue classes provides a method for reproducible extraction of this information. One problem that complicates this process, however, is intensity nonuniformity, an artifact in MR images which is evident as a gradual variation in intensity over otherwise identical tissue classes. Intensity nonuniformity has several causes, notably, inhomogeneity in radio frequency (RF) transmitter and receiver coils during image acquisition [1].

MR images provide excellent soft tissue contrast so that intensity-related features are natural choices for use with automatic segmentation methods. However, compensation for intensity nonuniformity must be included for such methods to be effective.

---

[1] A version of this chapter has been published. Withey et al. 2001. 23rd Int. Conf. IEEE EMBS 3:2657-2660.

Although it is possible to perform some compensation during image acquisition, equipment or protocol modifications are typically required. Furthermore, retrospective application of these corrective measures is not possible. Therefore, compensation applied as a post-processing step is considered to be desirable [2].

Adaptive, fuzzy c-means [2], [3], and statistically-based methods [4], [5] are examples of techniques which have been developed to perform automatic image segmentation in the presence of MR intensity nonuniformity. Other methods, such as nonlinear filtering [6], are intended to address the nonuniformity independently, permitting subsequent segmentation of the intensity corrected image.

Image segmentation can be performed by voxel labeling, involving classification of each image voxel or by identification of the bounding surfaces of objects in the image. The adaptive, fuzzy c-means methods [2], [3] and the statistical methods [4], [5] are examples of techniques which perform voxel labeling.

Determining the object boundaries in two-dimensional images can be done by application of active contours [10] or by edge tracing [11]. Here, a technique for edge tracing which includes a Kalman filter and a dynamic, target tracking algorithm to associate edge pixels into object boundaries is described.

3.2 Methodology

3.2.1 Synthetic Images

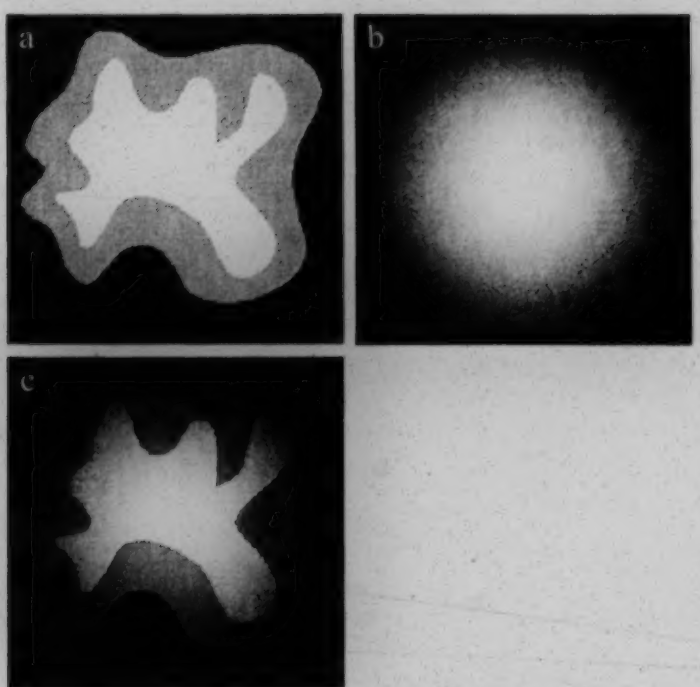

Figure 3-1. Synthetic Test Images.
(a) Unbiased image; (b) gain field; (c) biased image.

Figure 3-1 panels (a), and (c) show two synthetic test images. The shapes of the objects in the images have been chosen to resemble cortical gray matter and white matter in MR images of the brain. Each image has a size of 200x200 pixels with 256 gray levels. The unbiased image was formed by interpolating a small set of points with cubic splines to form boundaries of closed regions. These boundaries were then converted to discrete pixels and the enclosed regions were filled with a selected gray level value. The unbiased

test image has three gray levels with a difference of fifty gray levels between the brightest region and the intermediate intensity level.

In MR images, intensity nonuniformity has been approximated by exponential functions [1]. For the test images, the gain field (g), which simulates the intensity nonuniformity, was formed using a two-dimensional exponential function:

$$g = \exp(-k_1(X - X_c)^2 - k_2(Y - Y_c)^2) \tag{3-1}$$

where (X,Y) are pixel coordinates and (X_c,Y_c) are the coordinates of the image centre. Parameters k_1 and k_2 were chosen to provide a fifty percent intensity reduction at the image edges along the principal axes.

The biased image was formed by multiplying the unbiased image by the gain field, simulating an image with intensity nonuniformity.

3.2.2 Fuzzy c-Means Clustering

Fuzzy c-means clustering is a pattern recognition technique which is used in image segmentation [7]. Each pixel is evaluated according to a selected feature set, forming a set of vectors, or a set of points, in the feature space. Clusters are regions in the feature space with a high density of such points. A prototype vector, or cluster centre for each cluster is found by an iterative computation that minimizes the objective function

$$F = \sum_{k=1}^{N}\sum_{i=1}^{c} u_{ik}^{m} \times D_{ik}^{2} \tag{3-2}$$

where c is the number of clusters to form, N is the number of pixels in the image, m is the fuzzification factor (typically chosen to be 2), u_{ik} is the membership of pixel k in cluster i, $D_{ik} = \|f_k - v_i\|$ is the distance between the kth feature vector (f_k) and the

i th prototype vector (v_i). Once the prototype vectors and membership values have been found, each pixel can be assigned to the class of maximum membership to complete the segmentation.

Fuzzy c-means clustering is used here to demonstrate the problem that can occur when image segmentation is performed without attention to the effect of nonuniform intensity. The features selected for each pixel are the pixel intensity and the intensity of the four nearest neighbours. The number of clusters is three (i.e. $c = 3$) and the fuzzification factor is 2.

3.2.3 Dynamic Edge Tracing

Segmentation by edge tracing involves edge detection followed by association of edge pixels into object boundaries. In the case of the edge tracing method described here, edge detection is performed on the input image followed by a line-by-line scan of the edge image. On each line scan, the edge pixel positions and image intensity at the edge pixel are used as input to a multiple-target, dynamic, tracking algorithm. The scanning procedure introduces a history, allowing edges in the image to be followed along what amounts to a time dimension. Each new line brings a set of updated edge positions which are then associated with existing edge data from the previous lines. Edge positions that cannot be associated with an existing track are used to start two new tracks, one to the left and one to the right of the scan direction. Tracks which follow the same edge but in different directions will terminate on each other. These occurrences and the common start points are used to assemble the edges at the end of the scan.

The effective "movement" of the edge from one line to the next during the image scan simulates a dynamic system. Dynamic systems, linear or nonlinear, are described by state

and state transitions. State is a quantitative description of past and present behaviour, sufficient information to predict future behaviour. State transition is the description of how one state is transformed into another. For example, in aircraft tracking, state would include the position and velocity of the aircraft. The aircraft position could be predicted at a future time based on its current position and current velocity.

Automatic tracking algorithms are normally used to monitor the movement of aircraft or other targets of interest [8]. In the classical target tracking application, sensors provide measurements of the target state (e.g. position and velocity) to the tracking system at equal time intervals. The measured target data are compared to predicted target data and if sufficient correlation exists, the measured data are incorporated into the target history and a new prediction is formed for the next input sample. In this way, observations taken at different times can be associated together and the path taken by the target can be followed.

The functions of the tracking system are data association and state estimation. Data association is the process by which new data are correlated with existing data and the path of a target is updated. State estimation is the process whereby a target state estimate is computed using a priori noise statistics and past samples and whereby a predicted target state is determined. The target state estimate and next sample prediction are produced by a tracking filter with the prediction presented to the data association process at the next time interval. A block diagram of such a tracking system is shown in figure 3-2.

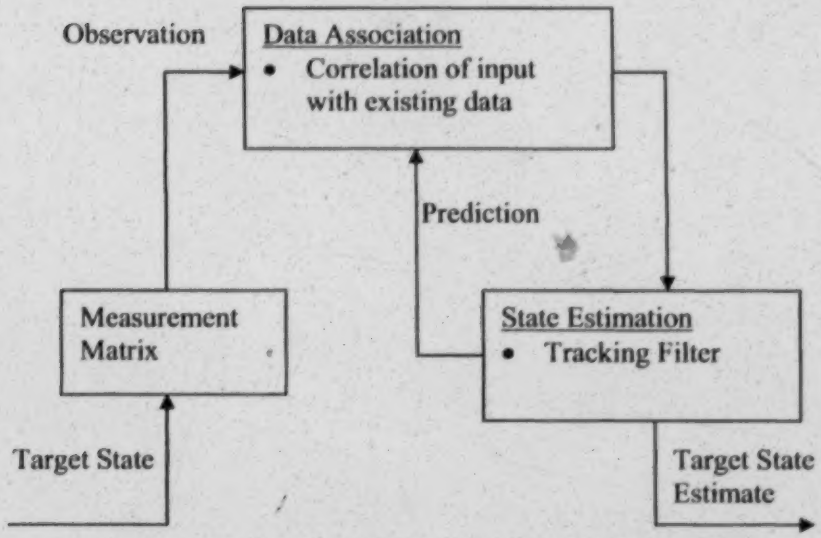

Figure 3-2. Tracking System Block Diagram.

In the edge tracing method described here, the tracking filter used for state estimation is a Kalman filter [9], the recursive solution to the discrete-time, linear, minimum variance estimation problem and the statistical estimator most often used in dynamic tracking [8]. For a given track, the Kalman filter is used to predict the edge position on the next line, facilitating the association of the next set of edge positions into the existing edge tracks.

The Kalman filter is defined with the assumptions of a linear, dynamic system and zero-mean, Gaussian noise. Gaussian distributions and linear dynamics are natural assumptions especially if statistical data is not largely available [9].

The Kalman filter can be used to estimate the state of a discrete process that is governed by the linear, stochastic difference equation

$$\mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \mathbf{w}_k \tag{3-3}$$

where k is the step counter, $\mathbf{x}_k$ is the state vector at step k, $\mathbf{A}_k$ is the state transition matrix, and $\mathbf{w}_k$ is the process noise vector. The process noise is assumed to be zero mean with Gaussian statistics. Measurements related to the target are also assumed to contain zero-mean, Gaussian noise:

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \tag{3-4}$$

where $\mathbf{z}_k$ is the measurement vector at step k, $\mathbf{H}_k$ is the measurement matrix, and $\mathbf{v}_k$ is the measurement noise vector.

The actual state of the target is not known and the Kalman filter is used to estimate it from the measurement and a previously determined state prediction. The estimate is taken to be a linear combination of the prediction and the difference between the measurement and the predicted measurement.

$$\hat{\mathbf{x}}_k = \tilde{\mathbf{x}}_k + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k \tilde{\mathbf{x}}_k) \tag{3-5}$$

where $\hat{\mathbf{x}}_k$ is the state estimate at step k, $\tilde{\mathbf{x}}_k$ is the state prediction at step k, and $\mathbf{K}_k$ is the Kalman filter gain. Also, the error covariance matrices are given by:

$$\mathbf{P}_k = E\{(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T\} \tag{3-6}$$

$$\tilde{\mathbf{P}}_k = E\{(\mathbf{x}_k - \tilde{\mathbf{x}}_k)(\mathbf{x}_k - \tilde{\mathbf{x}}_k)^T\} \tag{3-7}$$

where $\mathbf{P}_k$ is the a posteriori error covariance matrix, $\tilde{\mathbf{P}}_k$ is the a priori error covariance matrix, and $E\{\cdot\}$ represents mathematical expectation. The Kalman filter gain ($\mathbf{K}_k$) is determined by minimization of the trace of the error covariance matrix ($\mathbf{P}_k$) [9], [12].

At each measurement interval, the Kalman filter gain matrix, state estimate vector, and error covariance matrix are updated,

$$\mathbf{K}_k = \tilde{\mathbf{P}}_k \mathbf{H}_k^T (\mathbf{H}_k \tilde{\mathbf{P}}_k \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \tag{3-8}$$

$$\hat{\mathbf{x}}_k = \tilde{\mathbf{x}}_k + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \tilde{\mathbf{x}}_k) \tag{3-9}$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \tilde{\mathbf{P}}_k \tag{3-10}$$

where $\mathbf{R}_k = E\{\mathbf{v}_k \mathbf{v}_k^T\}$ is the measurement noise covariance matrix and $\mathbf{I}$ is the identity matrix.

Prediction of the next state is also done at each time interval

$$\tilde{\mathbf{x}}_{k+1} = \mathbf{A}_k \hat{\mathbf{x}}_k \tag{3-11}$$

$$\tilde{\mathbf{P}}_{k+1} = \mathbf{A}_k \mathbf{P}_k \mathbf{A}_k^T + \mathbf{Q}_k \tag{3-12}$$

where $\mathbf{Q}_k = E\{\mathbf{w}_k \mathbf{w}_k^T\}$ is the process noise covariance matrix.

The data association process will utilize the predicted error covariance matrix to form a bounding window around the predicted measurement. Any measurement that appears within this window is a candidate for association.

A simple two state filter is used where target state consists of position and velocity and the measurement is of position only. Under these conditions, $\mathbf{A}_k$ and $\mathbf{H}_k$ can easily be defined. That is, for

$$\mathbf{x}_k = [X, Y, \dot{X}, \dot{Y}]^T \tag{3-13}$$

where (X, Y) represents the target position in two dimensions and $(\dot{X}, \dot{Y})$ represents the target velocity in two dimensions and assuming a unit time step, the state transition and measurement matrices become:

$$\mathbf{A}_k = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3-14}$$

$$\mathbf{H}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{3-15}$$

which is to say that the next target state will be estimated from the current position and current velocity, that only position is measured, and that no particular measurement correction is required. Furthermore, these two matrices will remain constant for all k.

Our calculations are done in this manner with the exception that three dimensions are used, these being the two coordinates of the edge pixel and the image intensity at the edge pixel location.

3.3 Results

Fig. 3-3 shows the results from segmentation of the unbiased and biased test images by the edge tracing technique and by the fuzzy clustering approach. Panels (b) and (d) show that clustering works well when the intensity is uniform (b) but that given sufficient intensity nonuniformity, errors occur in the pixel assignments. In Panel (d), peripheral portions of the high intensity region are classed with lower intensity pixels and the central portion of the high intensity region is expanded, exhibiting a circularly-shaped artifact due to the exponentially-shaped intensity variation.

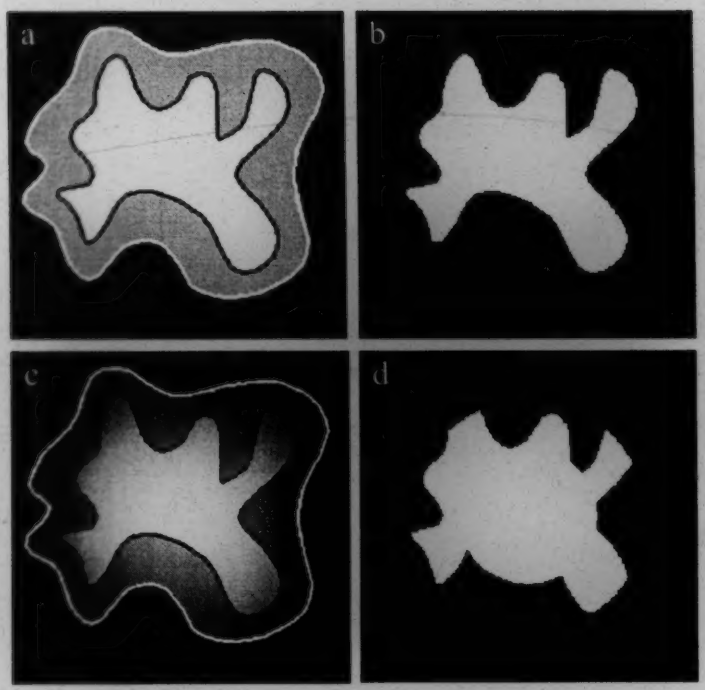

Figure 3-3. Segmentation Results.
(a) Edge tracing – unbiased image; (b) fuzzy clustering – unbiased image; (c) edge tracing – biased image; (d) fuzzy clustering – biased image.

The results from the dynamic edge tracing algorithm are shown in (a) and (c). In each case, the high intensity region is outlined with a black contour and the medium intensity region is outlined with a white contour. These lines coincide with the edge pixels very well.

The fit was evaluated by reconstructing the test image using each of the two sets of edge contours. Upon comparison with the original, unbiased test image, it was found that the reconstruction using contours from the unbiased image segmentation contained one pixel classified incorrectly. The reconstruction using contours from the biased image

segmentation contained ten misclassified pixels, amounting to 0.025 percent of all image pixels.

3.4 Discussion

Quantitative comparison of the fuzzy clustering result with the edge tracing result is not intended; however, having the two side by side gives an opportunity to consider the advantages of each. Although fuzzy clustering requires that the number of classes be known a priori, it can be extended to perform segmentation of three-dimensional (3D) images relatively easily. The edge tracing method does not require the number of tissue classes to be known a priori but is not as easily extended to 3D.

When 2D images are considered, operation without a priori knowledge of the number of tissue classes is a big advantage especially if the goal is automatic analysis of images where pathology may be involved. Fuzzy clustering may not position a cluster prototype so as to identify a relatively small region of distinct intensity in an image when there are much larger numbers of pixels in other intensity groups. Consideration of the objective function in (3-2) will confirm this. Since the objective function is based on minimizing the sum of all distances, a small but distinct group in the feature space may not have enough accumulated distance to attract a cluster centre. The edge tracing technique, however, would find all regions where there is an identifiable edge.

Edge tracing methods require some degree of edge continuity to be successful [11]. The edge tracing technique described here does not require adjacent pixel connectivity. Since the edge position is permitted to have a "velocity", the current velocity for that edge will determine where the algorithm searches for the next edge pixel.

3.5 Conclusions

Edge detection followed by a line-by-line scan of the edge image simulates a dynamic system where the edge position "moves" as the scan proceeds. The application of a dynamic tracking algorithm allows the edge to be followed, permitting edge pixels to be associated into object boundaries. Edge tracing performed in this manner and using three dimensions (the two edge pixel position coordinates and the image intensity at the edge pixel position) appears to be a viable method for 2D image segmentation in the presence of image intensity nonuniformity.

3.6 References

[1] B.R. Condon, J. Patterson, D. Wyper, A. Jenkins, D.M. Hadley, "Image non-uniformity in magnetic resonance imaging: its magnitude and methods for its correction," *British Journal of Radiology*, Vol. 60, 1987, pp. 83-87.

[2] S.K. Lee, M.W. Vannier, "Post-acquisition correction of MR inhomogeneities," *Magnetic Resonance in Medicine*, Vol. 36, 1996, pp. 275-286.

[3] D.L. Pham, J.L. Prince, "Adaptive fuzzy segmentation of magnetic resonance images," *IEEE Transactions on Medical Imaging*, Vol. 18, No. 9, 1999, pp. 737-752.

[4] K. Held, E. Rota Kops, B.J. Krause, W.M. Wells III, R. Kikinis, H.-W. Müller-Gärtner, "Markov Random Field Segmentation of Brain MR Images," *IEEE Transactions on Medical Imaging*, Vol. 16, No. 6, 1997, pp. 878-886.

[5] K. Van Leemput, F. Maes, D. Vandermeulen, P. Suetens, "Automated model-based bias field correction of MR images of the brain," *IEEE Transactions on Medical Imaging*, Vol. 18, No. 10, 1999, pp. 885-896.

[6] B.H. Brinkmann, A. Manduca, R.A. Robb, "Optimized homomorphic unsharp masking for MR grayscale inhomogeneity correction," *IEEE Transactions on Medical Imaging*, Vol. 17, No. 2, 1998, pp. 161-171.

[7] J.C. Bezdek, L.O. Hall, L.P. Clarke, "Review of MR image segmentation techniques using pattern recognition," *Medical Physics* Vol. 20, No. 4, 1993, pp. 1033-1048.

[8] E. Waltz, J. Llinas, "Multisensor Data Fusion," Artech House, 1990.

[9] R.E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME - Journal of Basic Engineering*, March, 1960, pp. 35-45.

[10] M. Kass, A. Witkin, D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, Vol. 1, 1988, pp. 321-331.

[11] M. Lineberry, "Image segmentation by edge tracing," in proc. of SPIE, The International Society for Optical Engineering, Vol. 359, *Applications of Digital Image Processing 4*, San Diego, Calif., USA, 1982, pp. 361-368.

[12] G. Welch, G. Bishop, "An Introduction to the Kalman Filter," *Report TR 95-041*, University of North Carolina at Chapel Hill, November, 2000.

Chapter 4

Comparison of Dynamic Edge Tracing and Classical Snakes

4.1 Introduction

Medical images such as magnetic resonance (MR) images and X-ray computed tomography (CT) images are typically held as two-dimensional (2D) arrays of picture elements (pixels) or three-dimensional (3D) arrays of volume elements (voxels, also called pixels). Segmentation of medical images, to identify specific objects or regions, can provide valuable information for diagnosis and treatment of disease. Manual segmentation of these images is possible but is a time consuming task subject to inter- and intra-operator variability so that automatic methods are preferable [1]. Automatic segmentation is an active research area and involves three main problems. First, the images contain noise that can alter the intensity of a pixel such that its classification becomes uncertain. Secondly, the images contain intensity nonuniformity where the average intensity level of pixels within a single tissue class varies over the extent of the image. And third, the images have finite pixel size and thus are subject to partial volume

averaging where individual pixels contain a mixture of tissue classes and the intensity of a pixel may not be consistent with any single tissue class.

There are two main approaches to automatic image segmentation. Pixel classification seeks to assign to each pixel in an image full or partial inclusion in a class or set of classes. This includes low-level techniques such as thresholds and region growing [2], [3] as well as the more sophisticated clustering methods based on concepts from pattern analysis [4]–[6]. Boundary identification, the subject of this study, seeks to identify the boundaries of objects or regions in an image and includes methods such as edge tracing [2], [3] as well as the more recently introduced active contours and active surfaces [4], [7], [8].

Simple edge tracing methods for 2D image segmentation are documented in textbooks on image processing [2], [3]. These algorithms typically involve edge detection and/or threshold operations to form a binary edge image followed by a tracing operation where pixels with adjacent neighbour connectivity are followed and collected into a list to represent the object boundary. Unfortunately, these algorithms can be very sensitive to noise and intensity nonuniformity that can cause gaps in the object boundary when a threshold is applied or can cause diversion away from the desired boundary during the tracing process. Despite these drawbacks, simple edge tracing continues to be applied when the nature of the problem permits or when augmented by further processing [9]–[11]. More advanced algorithms have also appeared [12]–[15], although these require substantial user input or else have limitations that restrict their ability to acquire contours in detailed, medical images, such as MR and CT head images.

A new edge tracing method where statistically-based, automatic target tracking algorithms are used in the edge tracing process to link edge pixels into object boundaries has been described in [16]. This approach has advantages over simple edge tracing since complete edge continuity is not required and since multiple edge features can be tracked simultaneously, permitting a higher level of discrimination in the tracing process without invoking complicated heuristics. In [16], a method using an automatic, line-by-line image scan was presented and applied to synthetic images. In the present work, edges are traced beginning at an operator-defined starting point to simplify the data processing and to focus on user-selected object boundaries. Intensity features obtained from regions adjacent to an edge are combined with edge location information to form the boundaries.

The purpose of this study is to evaluate this new algorithm in the context of the segmentation of MR and CT head images using the well known classical snakes algorithm for comparison. Both of these methods are 2D in nature, both are edge-based, both use primarily local information for determining an object boundary, and both are examples of image segmentation concepts in early stages of development.

The classical snake [7] is one of the first active contour algorithms and is intended for operation in 2D images. It is an artificial, closed contour modeled as a physical object with elasticity and rigidity parameters and is introduced into an image by a human operator. Edges (intensity gradients) within the image are modeled as external forces acting on the snake. During iterative update of an evolution equation, the snake moves to the nearest edge and conforms to it, identifying the boundary of an object within the image. The snake is affected very little by gaps in image edges since it is closed by definition; however, noise or spurious edges can interfere with its evolution and these

may be encountered at points relatively distant from the boundary of interest. Such occurrences are usually overcome by manual operator interaction. Other problems that have been identified include the need for the human operator to locate the initial contour very close to the desired edge, the difficulty of automatically expanding into narrow concavities, and the lack of edge selectivity for discrimination of closely spaced edges.

Despite these significant problems, development of snake-like algorithms has been undertaken with vigour and has produced a number of significant improvements [4], [8]. It is shown here that the dynamic edge tracing algorithm has advantages over the classical snakes algorithm in edge selectivity, entry into narrow concavities, execution time, and also can provide a mechanism for extracting multiple segmentation interpretations based on parameter selection. These advantages as well as parallels in the challenges faced by both methods suggest that further development of the dynamic edge tracing algorithm is possible and may produce a segmentation algorithm with unique capabilities.

In the following, section 4.2 provides the methods in two subsections, a review of classical snakes and a description of the new dynamic edge tracing algorithm. Section 4.3 contains results from the application of these methods to 2D MR and CT head images in situations that pose problems for classical snakes. Section 4.4 provides further discussion and section 4.5, the conclusions.

4.2 Methodology

4.2.1 Classical Snakes

The classical snakes algorithm is described in several publications, such as [7] and [8]. The snake is a thin, flexible, closed, 2D contour defined in a plane, $(x,y) \in \Re^2$, $\Re$ the

set of real numbers, and within a potential field, also defined on $\Re^2$, which can exert a force on the snake and deform it. The energy of the snake,

$$E(v) = S(v) + P(v) \tag{4-1}$$

is the sum of its internal energy, $S(v)$, and the energy supplied by the external field, $P(v)$, where $v = [x(s), y(s)]^T$ are the parametric coordinates of the snake and $s \in [0,1]$ is the parametric domain. The internal energy is given by

$$S(v) = \int_0^1 \left[\alpha(s) \left| \frac{\partial v}{\partial s} \right|^2 + \beta(s) \left| \frac{\partial^2 v}{\partial s^2} \right|^2 \right] ds$$

where $\alpha(s)$ is the snake elasticity and $\beta(s)$, the rigidity. The external energy is

$$P(v) = \int_0^1 p(v)\, ds$$

where $p(v)$ is the potential field in the vicinity of the snake.

The calculus of variations is used to determine that a minimum in $E(v)$ occurs when

$$-\frac{\partial}{\partial s}\left(\alpha(s) \frac{\partial v}{\partial s} \right) + \frac{\partial^2}{\partial s^2}\left(\beta(s) \frac{\partial^2 v}{\partial s^2} \right) + \nabla p(v) = 0. \tag{4-2}$$

Equation (4-2) can be interpreted as a force balance. The first two terms correspond to internal stretching and bending forces, respectively, and the third term corresponds to the external force.

Local minima of $E(v)$ are found by formulating a dynamic system based on Lagrangian mechanics, allowing the snake to move, expand, or contract within the potential field as a function of time. The Lagrange equations of motion are

$$\frac{\partial^2 \mathbf{v}}{\partial t^2}+\gamma\frac{\partial \mathbf{v}}{\partial t}+\left[-\frac{\partial}{\partial s}\left(\alpha(s)\frac{\partial \mathbf{v}}{\partial s}\right)+\frac{\partial^2}{\partial s^2}\left(\beta(s)\frac{\partial^2 \mathbf{v}}{\partial s^2}\right)\right]=-\nabla p(\mathbf{v},t) \qquad (4\text{-}3)$$

where $\mathbf{v}=\mathbf{v}(s,t)$ is also a function of time and γ is a step size constant. When the local minimum is reached, the time derivative will vanish and $\mathbf{v}(s)$ will be a solution to (4-2).

Application to a 2D image, $f(x,y)$, can be done by determining the potential field, $p(x,y)$, from the gradient of the image data so that the snake will be attracted to image edges,

$$p(x,y)=-\left|\nabla[G_\sigma * f(x,y)]\right|^2$$

where G_σ is a Gaussian filter with characteristic width, σ, that provides smoothing to limit the effect of noise and spurious edges. Implementation of (4-3) is by numerical methods, as detailed in [7].

4.2.2 Dynamic Edge Tracing

4.2.2.1 Dynamic Systems and Target Tracking

Dynamic systems are described by state and state transitions. State is a quantitative description of past and present behavior sufficient to predict future behavior [17]. State transition is the description of how one state is transformed into another. For example, in aircraft tracking, state would include the position and velocity of the aircraft. The aircraft state at future times could be predicted given knowledge of these variables and of the associated kinematics.

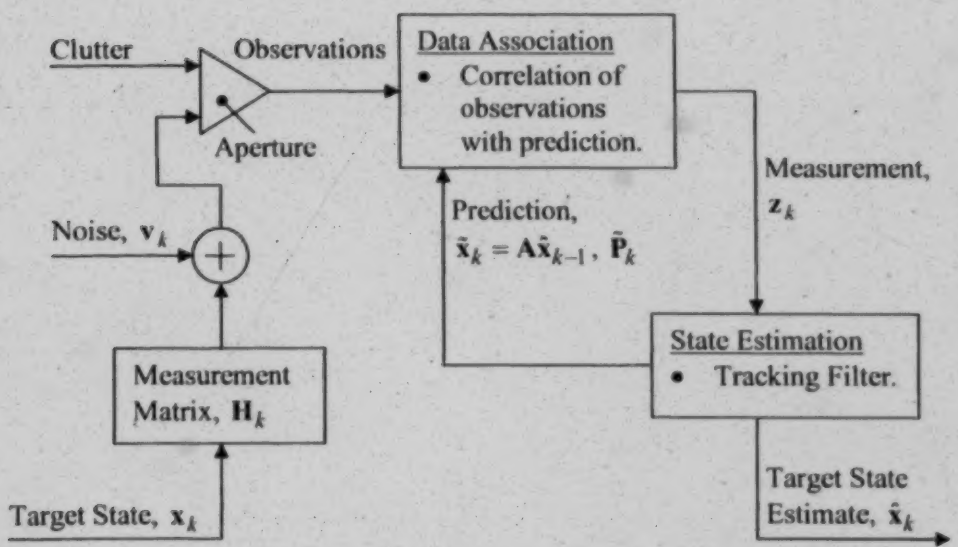

Figure 4-1. Tracking System Block Diagram

Automatic tracking algorithms are normally used to monitor the movement of targets of interest in 3D space [18]. In the classical target tracking application, sensors provide measurements of the target state to the tracking system at equal time intervals. The measured target data are compared to predicted target data and if sufficient correlation exists, the measured data are incorporated into the target history and a new prediction is formed for the next input sample. In this way, observations taken at different times can be associated together and the path taken by the target can be followed.

The block diagram of a tracking system is shown in figure 4-1. The true target state, $\mathbf{x}_k$, shown as an input to the system, cannot be measured directly. Instead, measurement of some of the target state variables can be made through the measurement matrix, $\mathbf{H}_k'$.

These also include noise. The output of the system is a filtered estimate, $\hat{\mathbf{x}}_k$, of the target state.

The main functions of the tracking system are data association and state estimation. Data association correlates new data with existing track data to identify a new measurement. State estimation computes the target state estimate using the new measurement, a priori noise statistics, and past information and also determines a target state prediction. The target state estimate and state prediction are produced by a statistical estimator, or tracking filter, with the prediction presented to the data association process at the next time step.

The Kalman filter [17], is the recursive solution to the discrete-time, linear, minimum mean square error estimation problem and is the statistical estimator most often used in dynamic tracking [18]. It is defined with the assumptions of a linear, dynamic system and zero-mean, Gaussian noise.

The Kalman filter can be used to estimate the state of a discrete process that is governed by the linear, stochastic difference equation,

$$\mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \mathbf{w}_k \tag{4-4}$$

where k is the step counter, $\mathbf{x}_k$ is the true state vector at step k, $\mathbf{A}_k$ is the state transition matrix, and $\mathbf{w}_k$ is the process noise vector. The process noise is assumed to have zero mean and Gaussian distribution. Measurements related to the target of interest are also assumed to contain zero-mean, Gaussian noise,

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \tag{4-5}$$

where $\mathbf{z}_k$ is the measurement vector at step k, $\mathbf{H}_k$ is the measurement matrix, and $\mathbf{v}_k$ is the measurement noise vector.

The true target state, $\mathbf{x}_k$, is not known and the Kalman filter is used to estimate it from $\mathbf{z}_k$ and a previously determined state prediction,

$$\hat{\mathbf{x}}_k = \tilde{\mathbf{x}}_k + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\tilde{\mathbf{x}}_k) \tag{4-6}$$

where $\hat{\mathbf{x}}_k$ is the state estimate at step k, $\tilde{\mathbf{x}}_k$ is the state prediction for step k, and $\mathbf{K}_k$ is the Kalman filter gain. The error covariance matrices are given by,

$$\mathbf{P}_k = E\{(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T\} \tag{4-7}$$

$$\tilde{\mathbf{P}}_k = E\{(\mathbf{x}_k - \tilde{\mathbf{x}}_k)(\mathbf{x}_k - \tilde{\mathbf{x}}_k)^T\} \tag{4-8}$$

where $\mathbf{P}_k$ is the a posteriori error covariance matrix, $\tilde{\mathbf{P}}_k$ is the a priori, or prediction, error covariance matrix, and $E\{\cdot\}$ represents mathematical expectation. The Kalman filter gain ($\mathbf{K}_k$) is determined by minimization of the trace of $\mathbf{P}_k$ [17], [19].

At each measurement interval the Kalman filter gain matrix, state estimate vector, and error covariance matrix are updated,

$$\mathbf{K}_k = \tilde{\mathbf{P}}_k\mathbf{H}_k^T(\mathbf{H}_k\tilde{\mathbf{P}}_k\mathbf{H}_k^T + \mathbf{R}_k)^{-1} \tag{4-9}$$

$$\hat{\mathbf{x}}_k = \tilde{\mathbf{x}}_k + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\tilde{\mathbf{x}}_k) \tag{4-10}$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\tilde{\mathbf{P}}_k \tag{4-11}$$

where $\mathbf{R}_k = E\{\mathbf{v}_k\mathbf{v}_k^T\}$ is the measurement noise covariance matrix and $\mathbf{I}$ is the identity matrix.

Prediction of the next state is also done at each time interval,

$$\tilde{\mathbf{x}}_{k+1} = \mathbf{A}_k \hat{\mathbf{x}}_k \tag{4-12}$$

$$\tilde{\mathbf{P}}_{k+1} = \mathbf{A}_k \mathbf{P}_k \mathbf{A}_k^T + \mathbf{Q}_k \tag{4-13}$$

where $\mathbf{Q}_k = E\{\mathbf{w}_k \mathbf{w}_k^T\}$ is the process noise covariance matrix.

Other items of note are the innovation, $\mathbf{y}_k$, defined as the difference between the measurement and the predicted measurement,

$$\mathbf{y}_k = \mathbf{z}_k - \mathbf{H}_k \tilde{\mathbf{x}}_k \tag{4-14}$$

and its covariance matrix, which appears in (4-9),

$$\mathbf{S}_k = \mathbf{H}_k \tilde{\mathbf{P}}_k \mathbf{H}_k^T + \mathbf{R}_k \tag{4-15}$$

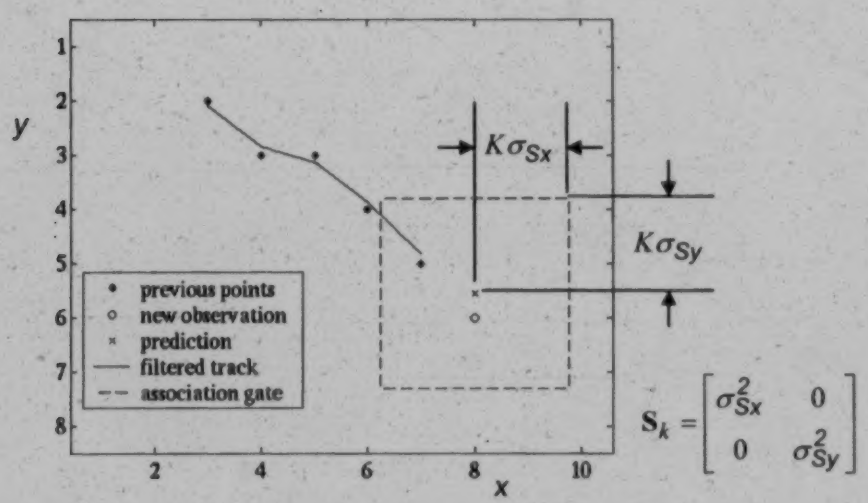

Figure 4-2. Example of Data Association.
Previous track measurements are shown as '*', the predicted point is shown as 'x' and the next observation as 'o'. The solid line is the filtered track. The association gate is shown as a dashed rectangle centered on the predicted point.

A set of measured data is obtained at each time interval. The data may include a measurement of the target location but may also include clutter, spurious measurements due to noise or other objects in the same vicinity. The data association process determines which, if any, new measurement can be associated with previous measurements to propagate the track and follow the target. This is done by placing an association gate around the prediction in the measurement space. The gate is sized using information from the elements of the innovation covariance matrix, $\mathbf{S}_k$. Figure 4-2 shows an example of a rectangular association gate in two dimensions. Other gate shapes, (e.g. circular or elliptical) are also possible. Parameter K is a constant used to set the number of standard deviations to be included in sizing the gate, with $K \cong 3$ a typical choice.

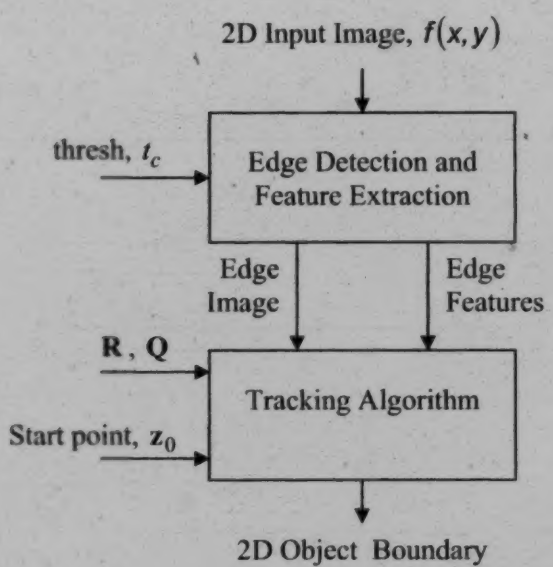

Figure 4-3. Processing Steps

4.2.2.2 Application to 2D Edge Tracing

Figure 4-3 shows a block diagram of the major processing steps in this approach to edge tracing. The raw data for the tracking algorithm is derived from the input image by edge detection and edge feature extraction. The edge information is used to model multidimensional measurements of the position of a hypothetical target obtained at regular time intervals as the target moved along the edge. The edge information includes data from the desired edge as well as nontarget data analogous to clutter in a typical target tracking scenario.

Beginning at an operator-defined starting pixel, the tracking algorithm follows a single edge until the starting point is revisited (i.e., a closed boundary is formed) or until no further pixels can be assigned to the track. In the event that the track terminates before locating the starting point, a second track is propagated beginning at the starting point and progressing in a direction opposite to that of the first track.

4.2.2.2.1 Edge Detection and Feature Extraction

A Canny edge operator [20] is used to produce a binary edge image. Canny operators are directional and combine low error rate with very good localization. Operators useful for step edges can be applied by computing directional derivatives of a Gaussian filtered image.

For $(x,y)\in\Re^2$, the image plane, and $f(x,y)$, the image, a Gaussian filtered image can be formed

$$f_{\sigma_c}(x,y) = G_{\sigma_c} * f(x,y)$$

where G_{σ_c} is a 2D, Gaussian filter with characteristic width, σ_c. Edge points are identified as zero crossings in the second derivative. These are further validated by the application of a threshold, t_c, to the first derivative to remove edges due to noise. The operator direction, θ, takes on six values, θ_j, with equal, 30 degree intervals. A binary edge image can be formed for each,

$$f_{bj}(x,y) = \begin{cases} 1, & \left|\frac{\partial}{\partial \boldsymbol{n}_j} f_{\sigma_c}(x,y)\right| \geq t_c \text{ and } \frac{\partial^2}{\partial \boldsymbol{n}_j^2} f_{\sigma_c}(x,y) = 0 \\ 0, & \text{otherwise} \end{cases}$$

where variable $\boldsymbol{n}_j$ is a unit vector in the operator direction, θ_j. In the discrete domain, edge pixels are those pixels that contain such validated edge points. The final edge image is the union of edge pixels determined for each operator direction.

Features related to the edge pixels can also be extracted during the edge detection process. Features of interest include the intensity of $f_{\sigma_c}(x,y)$ on each side of the edge, corresponding to intensities at the top and bottom of the intensity slope for a given edge pixel, referred to as the top and bottom intensities. This feature extraction is performed for each edge pixel and in each operator direction. Thus, each edge pixel has a two-element intensity feature vector associated with it. Edge pixels with zero crossings at more than one value of θ generate multiple instances of that edge pixel, each with a uniquely determined set of edge features.

4.2.2.2.2 Tracking Algorithm

Although there is no explicit time dimension, the time, Δ, between successive edge position measurements must be assigned. This is defined to be unity ($\Delta = 1$). In cases where the distances between edge pixels are not equal, the velocity of the hypothetical

target will change to compensate, resulting in target acceleration. Acceleration from pixel to pixel along an edge is modeled as white noise, reflected in the selection of the process noise covariance matrix, $\mathbf{Q}$.

A relatively simple Kalman filter is used where $\hat{\mathbf{x}}_k$ consists of components for position and velocity and $\mathbf{z}_k$ is a measurement of position only. Then,

$$\hat{\mathbf{x}}_k = \begin{bmatrix} \mathbf{p}_k \\ \Delta\mathbf{p}_k \end{bmatrix}, \tag{4-16}$$

where $\mathbf{p}_k$ is a vector of position and $\Delta\mathbf{p}_k$ is a vector of velocity, or position change, with element order corresponding to that in $\mathbf{p}_k$. With spatial and intensity features included, $\mathbf{p}_k = [p_{Xk} \quad p_{Yk} \quad p_{Tk} \quad p_{Bk}]^T$, where, the variables p_{Xk}, p_{Yk} represent the spatial dimensions and p_{Tk}, p_{Bk}, the intensity dimensions. Vector $\Delta\mathbf{p}_k$ contains the velocity estimate for each of these dimensions. Measurement $\mathbf{z}_k$ is in the same space as $\mathbf{p}_k$, $\mathbf{z}_k = [X_k \quad Y_k \quad T_k \quad B_k]^T$, with $T_k > B_k$.

Matrices, $\mathbf{A}_k = \mathbf{A}$ and $\mathbf{H}_k = \mathbf{H}$, will remain constant for all k. In two dimensions,

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4\text{-}17), \quad \mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{4-18}$$

which is to say that the state transition will be determined from the current position and current velocity (from the definition of $\mathbf{A}$) and that only position is measured (from $\mathbf{H}$). The patterns in each of these matrices will be retained when extended to four dimensions.

For proper initialization, $\mathbf{R}$, $\mathbf{Q}$, $\hat{\mathbf{x}}_1$, and $\mathbf{P}_1$ must be defined. The noise is assumed to have zero mean and constant variance so $\mathbf{R}_k = \mathbf{R}$ and $\mathbf{Q}_k = \mathbf{Q}$ are constant for all k. Matrix, $\mathbf{R}$, is used to specify the amount of measurement noise in the image. It is assumed that noise in each measurement dimension is independent of the others, producing a diagonal matrix,

$$\mathbf{R} = \begin{bmatrix} \sigma_X^2 & 0 & 0 & 0 \\ 0 & \sigma_Y^2 & 0 & 0 \\ 0 & 0 & \sigma_T^2 & 0 \\ 0 & 0 & 0 & \sigma_B^2 \end{bmatrix}. \tag{4-19}$$

The value of the noise variance in the spatial dimensions reflects the ability of the edge detector to locate edge pixels on the true edge in the image.

Use of the Kalman filter permits the target state to have a random component that can be used to model target maneuvers. A target dynamic model that assumes constant velocity with white noise acceleration is used. The process noise covariance matrix matching this target dynamic model is given by Blackman and Popoli [21] for the one-dimensional, two-state (position and velocity) case,

$$\mathbf{Q} = q \begin{bmatrix} \frac{\Delta^3}{3} & \frac{\Delta^2}{2} \\ \frac{\Delta^2}{2} & \Delta \end{bmatrix} \tag{4-20}$$

where q is a constant related to the acceleration statistics, and is determined by a tuning procedure in a typical tracking system. Expanding (4-20) to four dimensions, with Δ defined to be unity,

$$\mathbf{Q} = \begin{bmatrix} \frac{\mathbf{q}_m}{3} & \frac{\mathbf{q}_m}{2} \\ \frac{\mathbf{q}_m}{2} & \mathbf{q}_m \end{bmatrix}, \quad \mathbf{q}_m = \begin{bmatrix} q & 0 & 0 & 0 \\ 0 & q & 0 & 0 \\ 0 & 0 & q_T & 0 \\ 0 & 0 & 0 & q_B \end{bmatrix}. \tag{4-21}$$

Constant, q, applies to the spatial variables and q_T and q_B to the top and bottom intensity variables, respectively.

The position and velocity components of $\hat{\mathbf{x}}_1$ are determined from two edge pixel position measurements, $\mathbf{z}_0$ and $\mathbf{z}_1$. Measurement $\mathbf{z}_0$ is the operator-defined starting point and $\mathbf{z}_1$ is determined as the closest edge pixel to $\mathbf{z}_0$ within a default association gate centered on $\mathbf{z}_0$. State estimate, $\hat{\mathbf{x}}_1$, is then initialized with,

$$\mathbf{p}_1 = \begin{bmatrix} X_1 \\ Y_1 \\ (T_0 + T_1)/2 \\ (B_0 + B_1)/2 \end{bmatrix}, \quad \Delta\mathbf{p}_1 = \begin{bmatrix} X_1 - X_0 \\ Y_1 - Y_0 \\ 0 \\ 0 \end{bmatrix}. \tag{4-22}$$

The spatial dimensions are initialized as for a moving target whereas the intensity dimensions, which are not expected to be in transition at the point of track initialization and which typically have much higher noise variance values, are initialized as static.

Parameter $\mathbf{P}_1$ is determined following from [22],

$$\mathbf{P}_1 = \begin{bmatrix} \mathbf{R} & \mathbf{R} \\ \mathbf{R} & 2\mathbf{R} \end{bmatrix}. \tag{4-23}$$

In a typical tracking application, the measurements are taken at regular time intervals and, as a result, arrive in sequence. Edge pixels are not obtained in this way and sequencing must be performed. As shown in figure 4-4, this is done during data

association using distance measures within an elongated association gate. All pixels, $\mathbf{o}_i$, within the association gate at step k are examined according to the distance measures:

$$\breve{d}_i^2 = \mathbf{u}_i^T \mathbf{G}_k^{-1} \mathbf{u}_i \tag{4-24}$$

$$\hat{d}_i^2 = \mathbf{t}_i^T \mathbf{S}_k^{-1} \mathbf{t}_i \tag{4-25}$$

$$d_i^2 = \hat{d}_i^2 + \breve{d}_i^2 \tag{4-26}$$

with $\mathbf{t}_i = \mathbf{o}_i - \mathbf{H}\tilde{\mathbf{x}}_k$, $\mathbf{u}_i = \mathbf{o}_i - \mathbf{H}\hat{\mathbf{x}}_{k-1}$, and $\mathbf{G}_k = \mathbf{H}\mathbf{P}_{k-1}\mathbf{H}^T + \mathbf{R}$.

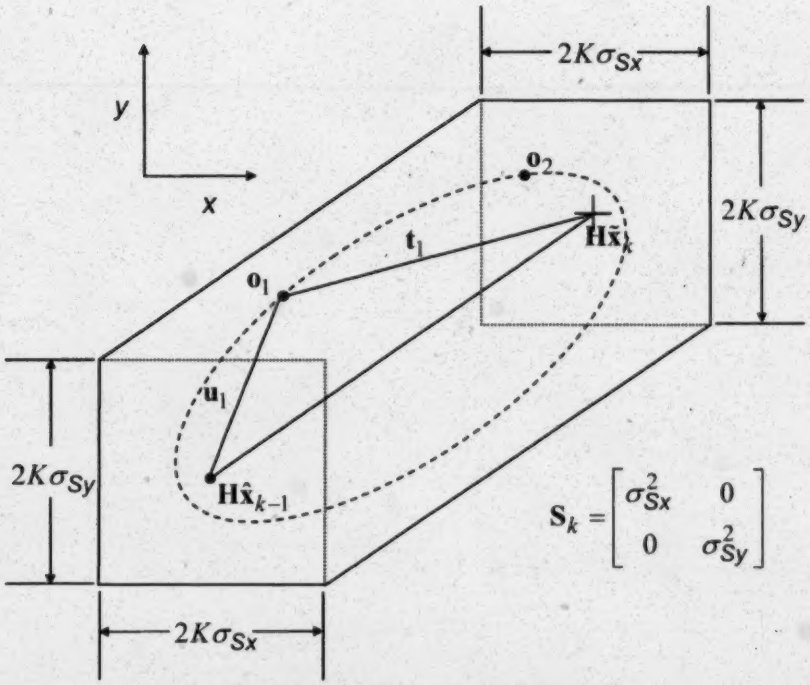

Figure 4-4. Distance Measure.

An elongated association gate with predicted position, $\mathbf{H}\tilde{\mathbf{x}}_k$, last filtered position, $\mathbf{H}\hat{\mathbf{x}}_{k-1}$, observations, $\mathbf{o}_1$, $\mathbf{o}_2$, and distances to $\mathbf{o}_1$, $\mathbf{u}_1$ and $\mathbf{t}_1$. A distance measure using the sum of $\mathbf{u}_1$ and $\mathbf{t}_1$ forms an ellipse upon which points with equal distance lie. Point sequencing is performed using a two step process: 1) find minimum distance; 2) of the points with the minimum distance, select the one closest to $\mathbf{H}\hat{\mathbf{x}}_{k-1}$.

For pixels that correspond to the minimum value of d_i^2, the pixel for which $\tilde{d}_i^2$ is minimum is taken as the next pixel in the track. This amounts to the selection of the closest pixel in the direction of the line between the state estimate and the prediction. Those pixels within the gate that are not selected by this process remain available for subsequent inclusion.

Two verification steps are also performed before a pixel is accepted for extension of the track. The first occurs because there are two possible directions for a track to proceed along an edge and because a reversal of direction is not desirable. Thus, the new data point must produce a track velocity vector that is consistent with the track direction. The second is needed to validate any jumps made by a track. A gap between pixels on an edge may be due to a minor intensity variation along the edge, or may separate two different, but possibly similar, edges. Since it is not desirable to have the track jump from one edge to another, interpolation of intensity data is performed across the gap, along the presumed top and bottom of the edge intensity slope. To be considered acceptable, the magnitude of the intensity change must not exceed the height of the edge, $(T_{k-1} - B_{k-1})$, along either of these paths.

Figure 4-5 shows an example image fragment and the edges that can be formed using edge detection with threshold, as described. Edge pixels are shown in gray in figure 4-5b. Figure 4-5c has two example tracks showing boundary selection: 1) when edge pixels are few so that gaps exist; and, 2) when thick edges occur due to multiple gradient directions.

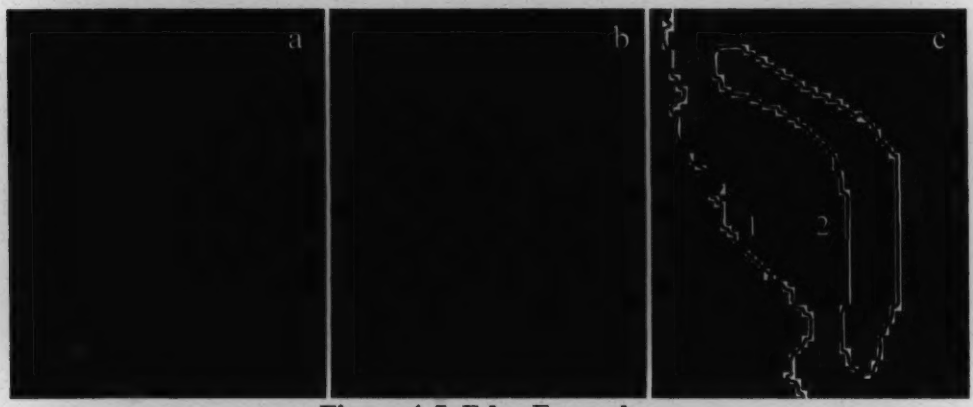

Figure 4-5. Edge Examples.

(a) 2D MR image fragment; (b) edge image with $t_c = 0.05$; (c) two example tracks showing gap jumping and operation in thick edges. Dark arrows show starting point locations for each track. White pixels are those chosen by the track. The black line in each case is the state estimate.

4.3 Results

Examples from a synthetic MR image, a real MR image, and a real CT image are described and the dynamic edge tracing results are compared to those for the classical snake. For edge tracing, the boundary identified by the edge tracing method is taken as the Kalman filter state estimate, $\hat{\mathbf{x}}_k$, $k = 1..n$, for a track consisting of n edge pixels. In each case, parameters $K = 3.38$ and $\sigma_c = 1$ were used. Low values of σ_c give better edge pixel localization but are more likely to produce multiple responses to a single edge. Threshold parameter, t_c, is given as a fraction of the maximum gradient magnitude.

For the snake, a value of $\sigma = 1$ was also used. Other snake parameters held constant were $\beta(s) = \beta = 0$ to allow the snake to form discontinuities and $\gamma = 1$ for unit time step in the equations of motion, (4-3). Parameter, $\alpha(s) = \alpha$, was set to a relatively low value in each example to promote the entry of the snake into boundary concavities, yet high

enough to maintain a smooth contour. Snakes were initialized very close to the edge of interest to avoid spurious edges and were interpolated to half pixel resolution. The snake deformation was limited to a fixed number of iterations and terminated once the contour had stabilized.

4.3.1 Synthetic MR Image

Slice number 79 from the brainweb synthetic MR image database [23]-[25] (normal, T1 weighted, 1mm thickness, 3% noise, and 20% intensity nonuniformity) is shown in figure 4-6a with area of interest identified. The four significant dark-colored regions within the area of interest are classed as containing mostly cerebrospinal fluid (CSF). Boundaries of these four regions were determined using the classical snake and using the edge tracing method.

Figures 4-6c and 4-6d show the snake and edge tracing results, respectively, overlaid on the relevant ground truth information from the synthetic image database with the dark pixels representing those where CSF was the dominant tissue class (CSF pixels). Two pixels of mostly gray matter at the lower extremity of region 3 and surrounded by CSF pixels were reclassified to CSF to facilitate the comparison.

The contours were evaluated based on the areas of CSF and non-CSF included in each region. Non-CSF pixels were defined as those in the synthetic image database having no CSF content. A third category, pixels with minor CSF content, was not evaluated. Ideally, all CSF pixels should fall within the boundaries and the non-CSF area should be zero. For comparison, the contours were discretized and CSF pixels on the discretized boundary were included in the region. Table 4-1 shows the evaluation data from the aggregate of the four regions. Although the differences are numerically small, they arise because the

edge tracing method is better able to adhere to the edge around sharp corners in the boundary.

Parameter selections for the edge tracing method are shown in Table 4-2 with $t_c = 0.1$. The value of the image noise variance after Gaussian filtering was approximately $\sigma_T^2 = \sigma_B^2 = 2$; however, the partial volume effect produced pixel-to-pixel intensity variations similar to random noise. This was modeled as higher measurement noise in the intensity dimensions.

Other parameters were selected for each region based on local considerations. A value of $q_B = 0.01$ in region 1 allowed the track to adhere to the edge along the lower extent of the region rather than crossing a thin projection, as does the snake. In region 3, the variations in intensity on the brighter side of the edge are less, and on the darker side greater, than those in the other regions, resulting in modified parameter selections. Snake parameter, $\alpha = 0.01$, was used, with 400 iterations permitted.

Table 4-1. Synthetic MR Image Comparison Data

Total CSF area = 1158 pixels.

Category	Snake	Edge Tracing
CSF pixels inside/on boundary	1131 (97.7 %)	1145 (98.9 %)
Non-CSF pixels inside/on boundary	100	95
Non-CSF pixels inside boundary	5	2

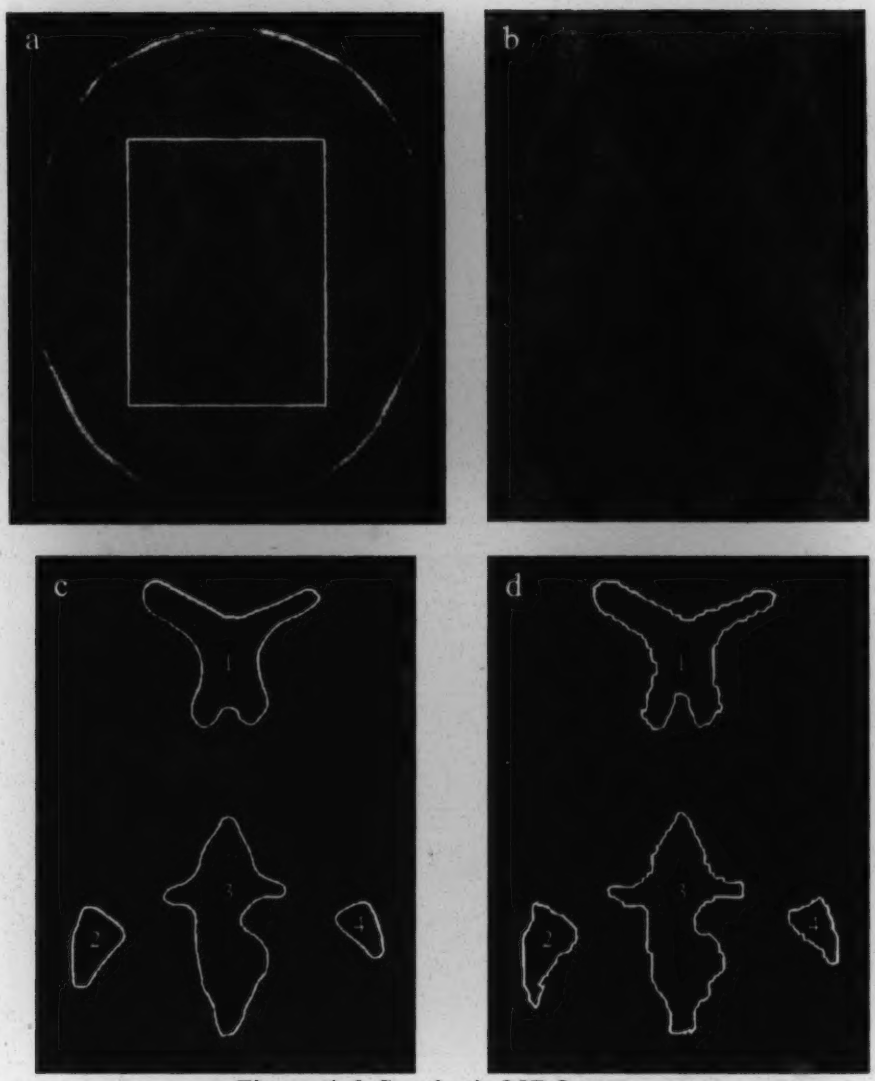

Figure 4-6. Synthetic MR Image.

(a) 2D MR slice image with area of interest identified; (b) area of interest. Four regions from the brain ventricles (CSF) were selected; (c) snake result (white line) overlaid on the known data. Black pixels are those where CSF is the dominant tissue type; (d) edge tracing result also overlaid on the known data. The tracing algorithm is able to adhere to the edge even around sharp curves in the boundary.

Table 4-2. Edge Tracing Parameters (Figure 4-6)

Region	Measurement Noise				Process Noise		
	σ_X^2	σ_Y^2	σ_T^2	σ_B^2	q	q_T	q_B
1	0.2	0.2	8	8	0.2	20	0.01
2	0.2	0.2	8	8	0.2	20	0.1
3	0.2	0.2	8	8	1	2	2
4	0.2	0.2	8	8	0.2	20	0.1

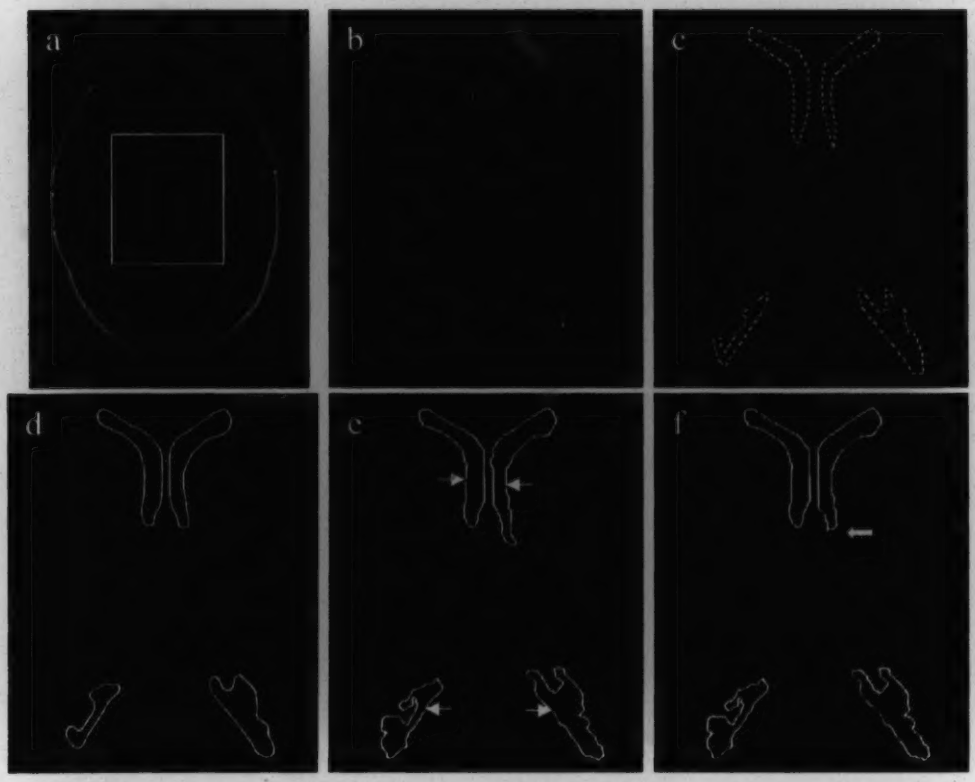

Figure 4-7. Real MR Image.
(a) 2D MR slice image with area of interest identified; (b) zoomed area with four regions identified; (c) initial snake contours; (d) final snake contours. Snake does not retain a small partial volume area at the lower tip of region 2 although it is included in the initial contour; (e) edge tracing result. White arrows show starting point locations for each region. The contour of region 2 absorbs the small partial volume area at the lower tip; (f) edge tracing result after parameter modification showing modified contour in region 2 (white arrow).

4.3.2 Real MR Image

Figure 4-7a shows a real MR image from which a fragment has been extracted and displayed in figure 4-7b. Four regions from the lateral ventricles were selected and the boundaries of each were identified using both methods. Figure 4-7c shows the initial contours for the classical snake. The final contours for each method are shown overlaid on the image in figures 4-7d and 4-7e. The edge tracing method appears to permit a greater level of detail in the resulting boundaries.

In region 2, the snake retracted from a small partial-volume extension although initialized to include it. The edge tracing algorithm is able to include or exclude this feature based on parameter selections. To capture it, $q_B = 2$ was used since a higher value of the dynamics parameter allows the track greater maneuverability in that dimension. Alternatively, as shown in figure 4-7f, setting $q_B = 0.01$ restricts the track in the B dimension and the partial volume extension can be avoided.

Other parameter selections for the edge tracing method are shown in Table 4-3 with $t_c = 0.05$. Regions 3 and 4 of figure 4-7b show lower contrast than regions 1 and 2 due to substantial partial volume effect. A variety of possible boundary interpretations exist. Higher noise variance in the intensity dimensions was chosen for these regions to model more of the data variation as measurement noise. This allows lower dynamics parameters to be used and can lead to a more stable track. Snake elasticity was $\alpha = 0.01$ and 400 iterations were permitted.

Table 4-3. Edge Tracing Parameters (Figure 4-7e)

Region	Measurement Noise				Process Noise		
	σ_X^2	σ_Y^2	σ_T^2	σ_B^2	q	q_T	q_B
1	0.2	0.2	10	10	0.2	2	0.01
2	0.2	0.2	10	10	0.2	2	2
3	0.2	0.2	20	20	0.2	0.2	0.2
4	0.2	0.2	20	20	0.05	0.2	0.2

As shown in figure 4-8, the selection of parameter, q, in the spatial dimensions can also allow multiple boundary interpretations to be realized. In figure 4-8b, with $q = 0.05$, the track descends along the left side of the dark colored area and then ascends along the right side but traces a wider boundary than is found in figure 4-8c, for which $q = 0.08$. This is consistent with the function of q where higher values suggest a higher probability of a maneuver in the spatial dimensions and cause a tighter adherence to the measurements. In this case, the parameter modification changes the velocity vector sufficiently at the critical point that a different path is taken. The sequence of pixels chosen by the track, however, also depends on the other parameters so that the choice of q influences the path selection but does not independently determine it.

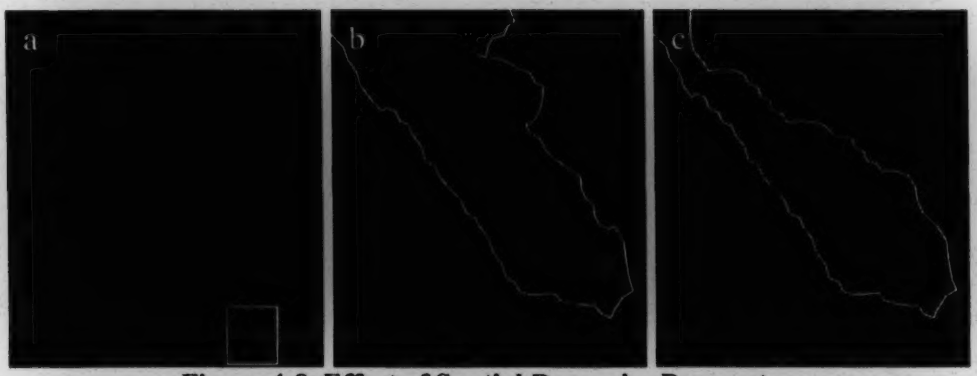

Figure 4-8. Effect of Spatial Dynamics Parameter.
(a) Area of interest; (b) q=0.05; (c) q=0.08.

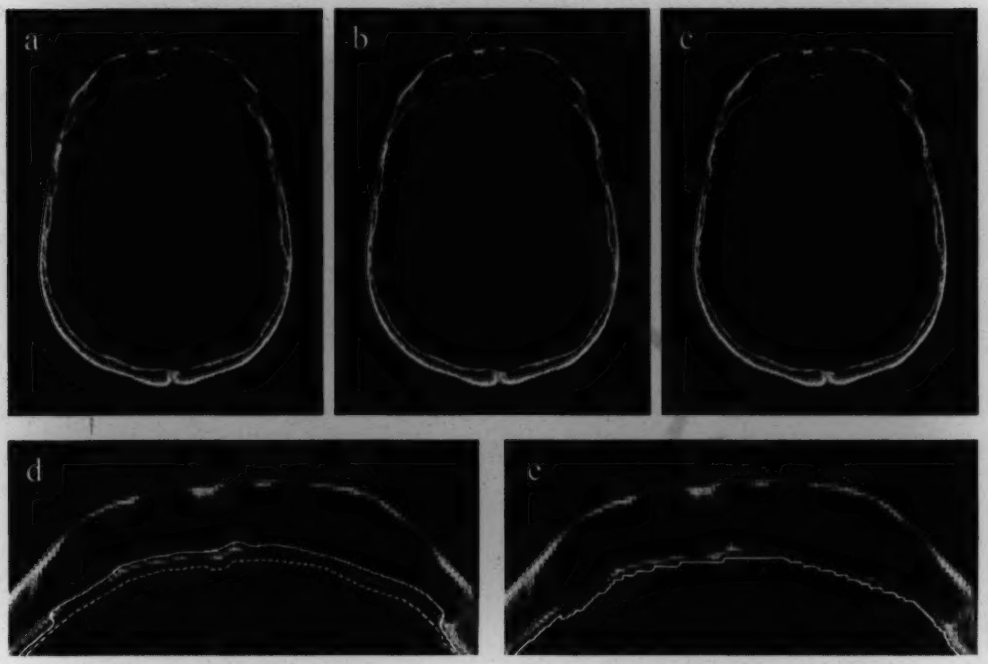

Figure 4-9. CT Image - Soft Tissue Boundary.
(a) 2D CT slice image. The edge of interest is the inner-skull/soft-tissue boundary; (b) snake result; (c) edge tracing result. Black arrow shows location of initial starting point; (d) zoomed snake result with snake shown in white for better contrast. The dotted line is the initial snake location. The snake is attracted to the higher gradient through thin sections of the bone region; (e) zoomed edge tracing result. The track remains on the soft tissue side of the boundary.

4.3.3 Real CT Image

Figure 4-9a shows CT image c_vm1072 from the United States National Library of Medicine Visible Human Project, cropped to 241x186 pixels and scaled to 256 gray levels. In figure 4-9b, the black line is the contour produced by the application of a classical snake to identify the soft tissue boundary inside the skull. This appears to be acceptable except along a very thin bone region next to the dark-colored, frontal sinuses at the top of the image. At the thinnest points along this feature, the snake was able to pass through the bone region and adhere to the higher-gradient, frontal sinus boundary. The remainder of the snake was then pulled beyond the desired edge by the higher force, aided by its own internal elasticity. A zoomed view is shown in figure 4-9d with the snake contour displayed in white for better contrast in the zoomed image. The dashed white line is the initial location of the snake. Many snake parameter selections were tried but no combination was found that could resolve this problem.

As shown in figure 4-9c, the edge tracing method is able to distinguish the desired edge and is not influenced by the stronger, nearby edge that captures the snake. However, there is one section at the left side of figure 4-9e where the track is slightly distanced from the true edge. This appears to be due in part to path selection in low-contrast conditions where the number of possible paths has increased and to poorer edge localization as can occur when the image edge does not form a good approximation to a step edge.

Edge tracing parameters for the images in figure 4-9 are given in Table 4-4 with $t_c = 0.01$. For the snake, $\alpha = 0.1$ with 800 iterations permitted.

Table 4-4. Edge Tracing Parameters (Figure 4-9)

Measurement Noise				Process Noise		
σ_X^2	σ_Y^2	σ_T^2	σ_B^2	q	q_T	q_B
0.1	0.1	100	5	0.2	1	0.1

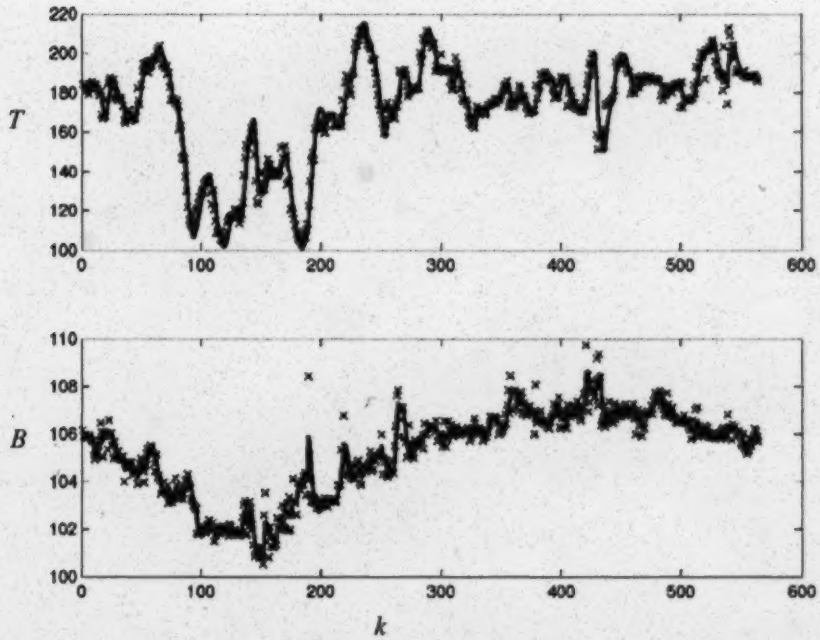

Figure 4-10. Intensity Features.

Intensity features for the edge tracing result of figure 4-9. In each plot, the crosses mark the measurements and the solid line is the Kalman filter state estimate. Significant variation, as much as 50% reduction at places relative to the peak, occurs in T with the lowest values found along the edge next to the frontal sinuses. Variation in B is considerably less, roughly 10%, and gradual intensity nonuniformity is clearly visible along the length of the track.

Figure 4-10 shows the intensity features, T_k and B_k, for the edge track of figure 4-9c, beginning at the starting point and for the full extent of the track. In each plot, the crosses mark the measurements and the solid line is the Kalman filter state estimate in the corresponding dimension. The top intensity, T_k, shows very significant variation, up to approximately 50% reduction relative to the peak, with the lowest values found along the edge next to the frontal sinuses. The variation of B_k is considerably less, roughly 10%. Gradual intensity nonuniformity is clearly visible in this track dimension.

4.3.4 Execution Time

These tests were conducted using MATLAB® [26] version 6.0 on a 250 MHz SUN Sparc Ultra 30 workstation. Average execution time for the edge tracing method was approximately 70 milliseconds per track sample. This time is dependent on the density of measurements in the association gate, affected by edge thickness and noise edge pixels, and the number of gaps encountered, where interpolation is required. The extraction of intensity features required about 1 millisecond per edge pixel.

Average execution time for the snake was 540 milliseconds per contour point, considerably longer than for the edge tracing method; however, 400 or more iterations were allowed to ensure convergence. In some cases the snake may have reached a shape reasonably close to the final contour in a shorter time although it was often difficult to determine whether convergence had occurred because the snake typically remained active throughout the test. Snake execution time depends on the initial contour placement and the number of points in the contour. On each iteration, an NxN pentadiagonal matrix must be inverted where N is the number of samples in the contour.

4.4 Discussion

A comparison with classical snakes may initially appear to be inconsequential because many improvements to the snakes algorithm have been developed since its introduction [8]. However, this comparison allows a perspective to be gained regarding the potential of the dynamic edge tracing algorithm since both algorithms represent an early stage in their respective development. In fact, many of the problems that affect classical snakes are problems also faced by dynamic edge tracing. These include limitation to 2D images, sensitivity to noise, sensitivity to starting configuration, possibility of self intersection, the selection of appropriate parameter values, and the requirement for operator intervention when a valid segmentation result is not formed. In the case of active contours, all but the last two of these have been resolved after more than a decade of subsequent research effort. This suggests that similar developments could be possible for dynamic edge tracing, given further research.

The advantages of the edge tracing algorithm in this comparison are edge selectivity, the ability to enter narrow concavities, a greater set of possible segmentation interpretations based on parameter selection, and lower execution time. The main advantage of the classical snakes algorithm is that closed contours are guaranteed, although this is a true advantage only in cases where a satisfactory segmentation is produced without the need for significant operator intervention.

For the edge tracer, there is no certainty that a closed contour will be formed in all cases and operator interaction may be required to adjust parameter values, to prune wayward tracks, or to initiate additional tracks so that the object of interest is completely enclosed. In the examples presented, operator actions were limited to parameter selection,

setting the starting contour for the snake, and selection of the starting point for edge tracing.

The edge tracing initialization requires selection of a single point on the desired boundary and thus contains no information about the final shape of the contour. Tracking success is improved when a starting point that is representative of others on the boundary, and that has similar neighbours, is chosen so that valid track initialization occurs.

For edge tracing parameter selection, rapid, sample-to-sample variations in target position are usually best modeled as measurement noise whereas multisample trends, such as contour curvature and intensity nonuniformity, are best modeled using the process noise dynamics parameters. Lower values of the dynamics parameters tend to produce tracks with greater stability.

The parameters for the edge tracing method are all related to the image statistics. Although promising, this does not necessarily relieve the problem of parameter selection and repeated trials with alternate parameter values, since local variations in image features may still require adjustment from the nominal values. The selection of the dynamics parameters, q, q_T, q_B, involves the acceleration statistics of the hypothetical target. These are not readily available a priori and the parameter values may need to be revised based on the results from a few initial trials.

The classical snake contour is much smoother than that produced by the edge tracing method. It may be possible to improve contour smoothness in the edge tracing result by applying a Kalman smoother [19] as a post-processing step.

In cases of very low contrast, the distance measure used by the edge tracing method for recursive pixel selection may not always select the path that best represents the

desired edge. Using gradient magnitude information in the pixel selection process may be one way to improve this.

The localization of the edge pixels is dependent on the edge detection operator. The Canny edge operator used here was originally developed for step edges and localization may suffer when the edge does not approximate a step.

Advanced tracking algorithms exist which evaluate multiple hypotheses and thus defer the selection of next measurement until after further track development [21]. Use of these may provide increased robustness in the selection and sequencing of pixels and may also permit the automatic formation of multiple segmentation interpretations, which could then be evaluated by higher-level processing. These benefits would be especially valuable in the context of unsupervised operation.

The Kalman filter is capable of operating with many more dimensions than shown here. This capability may also permit the use of multichannel data in the edge tracing process. The use of this data in edge-based segmentation has previously been identified as being a difficulty [27] although such data are available from multiecho MR imaging scans, for example, and are often used by the clustering methods in the segmentation of MR images.

Noise in both MR and CT images is often modeled using Gaussian distributions. In MR images the noise distribution is more accurately described as Rician. This is approximately Gaussian for signal to noise ratio (SNR) greater than 3 [28], where $SNR = A/\sigma_N$ is the ratio of the signal amplitude in the absence of noise to the noise standard deviation. In the image background, where there is no MR signal, the noise follows a Rayleigh distribution. MR images from clinical scanners very often exhibit

SNR well above 3 so the assumption of Gaussian noise distributions in these images is often valid. As an example of *SNR* value, although darkly colored, the CSF regions in the synthetic MR image of figures 4-6a and 4-6b have *SNR* of approximately 10.

4.5 Conclusion

An advanced edge tracing algorithm based on dynamic target tracking is capable of identifying boundaries in medical images in circumstances that pose difficulty for classical snakes. Specifically, entry into narrow concavities and in cases where edge specificity is required such as along relatively weak edges and along very thin image features that bound high gradients. These are cases where the physical model of the snake, the internal energy and the external forces, can adversely affect its ability to conform to the desired boundary. Also, this edge tracing algorithm exhibits relatively efficient execution times and multiple segmentation interpretations can be formed based simply on parameter selection.

The edge tracing algorithm is not dependent on adjacent pixel connectivity for continuation of the tracing process. A threshold is used to separate noise edges from object edges but relatively low thresholds are possible and thick edges can be accommodated. The Kalman filter, which forms the core of the tracking algorithm, can operate in multiple dimensions, permitting the seamless use of multiple features to improve discrimination of object boundaries.

Results from MR and CT images show that this edge tracing method is able to delineate boundaries in image regions containing noise, intensity nonuniformity, and partial volume effects. Its statistical foundation allows modeling of image noise and its recursive nature and process model allow it to follow edges through intensity

nonuniformity consisting of smoothly varying image acquisition artifact as well as more abrupt variations that can occur in cases of partial volume averaging.

These are valuable capabilities; however, some problems remain. As with other edge tracing algorithms, there is no certainty that closed contours will be formed. In addition, a number of parameters must be defined. Though related to image statistics, these parameters may need to be tuned to the local environment for successful application. Further work may allow these problems to be overcome.

4.6 References

[1] L.P. Clarke, R.P. Velthuizen, M.A. Camacho, J.J. Heine, M. Vaidyanathan, L.O. Hall, R.W. Thatcher, M.L. Silbiger, "MRI segmentation: Methods and applications," *Magnetic Resonance Imaging*, Vol. 13, No. 3, 1995, pp. 343-368.

[2] I. Pitas, "Digital Image Processing Algorithms," Prentice-Hall, 1993.

[3] A.R. Weeks. "Fundamentals of Electronic Image Processing," SPIE Optical Engineering Press, IEEE Press (copublishers), 1996.

[4] D. Pham, C. Xu, J. Prince, "Current methods in medical image segmentation," *Annual Review of Biomedical Engineering*, Vol. 2, 2000, pp. 315-337.

[5] D.L. Pham, J.L. Prince, "Adaptive fuzzy segmentation of magnetic resonance images," *IEEE Transactions on Medical Imaging*, Vol. 18, No. 9, 1999, pp. 737-752.

[6] Y. Zhang, M. Brady, S. Smith, "Segmentation of brain MR images through a hidden Markov random field model and the expectation maximization algorithm," *IEEE Transactions on Medical Imaging*, Vol. 20, No. 1, 2001, pp. 45-57.

[7] M. Kass, A. Witkin, D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, Vol. 1, No. 4, 1988, pp. 321-331.

[8] T. McInerney, D. Terzopoulos, "Deformable models in medical image analysis: A survey," *Medical Image Analysis*, Vol. 1, No. 2, 1996, pp. 91-108.

[9] J. Solomon, B. Widemann, K. Warren, F. Balis, N. Patronas, "Automatic lesion detection and volume measurement in MR imaging of plexiform neurofibromas," in: Proc. 2002 IEEE International Symposium on Biomedical Imaging, pp. 229-232.

[10] H. Soltanian-Zadeh, J.P. Windham, "A multiresolution approach for contour extraction from brain images," *Medical Physics*, Vol. 24, No. 12, 1997, pp. 1844-1853.

[11] M. Gordan, C. Kotropoulos, I. Pitas, "Pseudoautomatic lip contour detection based on edge direction patterns," in: Proc. 2nd International Symposium on Image and Signal Processing and Analysis, June 2001, pp. 138-143.

[12] M. Basseville, B. Espiau, J. Gasnier, "Edge detection using sequential methods for change in level – Part I: A sequential edge detection algorithm," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-29, No. 1, 1981, pp. 24-31.

[13] M. Lineberry, "Image segmentation by edge tracing," in: Proc. of SPIE, The International Society for Optical Engineering, Vol. 359, *Applications of Digital Image Processing IV*, San Diego, Calif., USA, 1982, pp. 361-368.

[14] A.X. Falcão, J.K. Udupa, F.K. Miyazawa, "An ultra-fast user-steered image segmentation paradigm: Live wire on the fly," *IEEE Transactions on Medical Imaging*, Vol. 19, No. 1, 2000, pp. 55-62.

[15] S. Mahamud, L.R. Williams, K.K. Thornber, K. Xu, "Segmentation of multiple salient closed contours from real images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 4, 2003, pp. 433-444.

[16] D.J. Withey, Z.J. Koles, W. Pedrycz, "Dynamic edge tracing for 2D image segmentation," in: Proc. 23rd Int. Conf. IEEE Engineering in Medicine and Biology Society, Vol. 3, Oct. 2001, pp. 2657-2660.

[17] R.E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME - Journal of Basic Engineering*, March, 1960, pp. 35-45.

[18] E. Waltz, J. Llinas, "Multisensor Data Fusion," Artech House, 1990.

[19] A. Gelb, ed. "Applied Optimal Estimation," MIT Press, Cambridge MA, 1974.

[20] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8, No. 6, 1986, pp. 679-698.

[21] S. Blackman, R. Popoli, "Design and Analysis of Modern Tracking Systems," Artech House, 1999.

[22] Y. Bar-Shalom, T.E. Fortmann, "Tracking and Data Association," Academic Press, 1988.

[23] http://www.bic.mni.mcgill.ca/brainweb/

[24] C.A. Cocosco, V. Kollokian, R.K.-S. Kwan, A.C. Evans, "BrainWeb: Online interface to a 3D MRI simulated brain database," *NeuroImage*, Vol. 5, No. 4, part 2/4, S425, 1997 -- Proceedings of 3rd International Conference on Functional Mapping of the Human Brain, Copenhagen, May 1997.

[25] R.K.-S. Kwan, A.C. Evans, G.B. Pike : "An extensible MRI simulator for post-processing evaluation," Visualization in Biomedical Computing (VBC'96). Lecture Notes in Computer Science, Vol. 1131. Springer-Verlag, 1996, pp. 135-140.

[26] http://www.themathworks.com

[27] S. Dellepiane, "Image segmentation: Errors, sensitivity, and uncertainty," in: *IEEE Engineering in Medicine and Biology Conference*, Vol. 13, No. 1, 1991, pp. 253-254.

[28] H. Gudbjartsson, S. Patz, "The Rician distribution of noisy MRI data," *Magnetic Resonance in Medicine,* Vol. 34, 1995, pp. 910-914.

Chapter 5

Dynamic Edge Tracing for Identification of Boundaries in Medical Images[2]

5.1 Introduction

In medical images, the identification of object boundaries or regions of interest can provide valuable information for diagnosis and treatment of disease. Such segmentation operations can be performed manually but are very time consuming and subject to inter- and intra-operator variability so that automatic methods are preferable [1].

Segmentation of medical images involves three main problems. The images contain noise that can alter the intensity of a pixel such that its classification becomes uncertain. The images contain intensity nonuniformity where the average intensity level of a single tissue class varies over the extent of the image. And, the images have finite pixel size and thus are subject to partial volume averaging where individual pixels contain a mixture of tissue classes and the intensity of a pixel may not be consistent with any one class.

---

[2] A version of this chapter has been submitted for publication. Withey et al 2005. Computer Vision and Image Understanding.

Automatic image segmentation is an area of active research that has produced a variety of methods. Historically, these could be divided into three main approaches: region-based methods, pixel classification, and edge detection, including edge tracing [2]. The same imprint can be seen today especially when the methods of active surfaces [3], [4] are viewed as advances in region growing [5]. Likewise, advances in pixel classification have produced sophisticated clustering algorithms [3], [6] but similar advances in automatic edge tracing have not been so apparent.

Automatic and semiautomatic edge tracing are distinguished by the level of involvement required by the operator. Graph search methods [7], [8], for example, are semiautomatic methods that require continuous input from the operator during the tracing process. Automatic edge tracing methods require operator initialization, which may include the identification of a seed point, but subsequent operator involvement is not required until after the tracing operation is complete.

The main criticism that has been directed toward automatic edge tracing as an approach to image segmentation has been poor robustness. However, the identification of a coherent boundary by linking neighbouring edge points provides useful information for the purpose of segmentation and is information not obtained by other methods.

The earliest edge tracing algorithms required 8-neighbour connectivity for successive edge pixels, and propagation from one pixel to the next was often performed solely on the basis of local gradient information [7]. These methods were very sensitive to noise and intensity nonuniformity that can cause discontinuities in the object boundary or induce deviation from the boundary during the tracing process. Efforts have been made to improve robustness by providing a means for crossing discontinuities and by

incorporating local intensity information for better edge selectivity [9]-[12]. However, local image information alone does not, in general, permit the production of high quality boundaries.

Algorithms that integrate local and global image information have also appeared. A Hopfield neural network was used to combine information from spatially-separated edge segments to form contours in a 2D color image [13]. This was tested on a limited dataset so it is not clear how performance would vary in cases of significant intensity noise or of irregularly-shaped contours. Multiresolution pyramids were used to connect discontinuities in traced edges for extraction of the inner and outer skull contours and the skin contour in medical head images [14] but a binary, intensity threshold was used which can cause problems in cases of intensity nonuniformity. A global saliency relation was developed by modeling the paths of hypothetical, constant-speed particles undergoing Brownian motion to link edge segments into object boundaries [15]; however, this method has been explicitly directed toward images with very smooth object contours. Size constraints and a centrally-located, seed pixel were used to obtain boundaries from ultrasound and computed tomography (CT) images [16], although certain contour convexity requirements exist for this method.

As with [16], although developed independently, the potential for automatic target tracking algorithms to serve as advanced edge tracers has been recognized in the development of Dynamic Edge Tracing, proposed as a new approach for medical image segmentation where edge tracing is guided by a dynamic system model [17]. Edge information in an image is modeled as position information obtained at regular time intervals from a hypothetical target moving along the edge. These position measurements

may also contain noise. The edge is traced, identifying an object boundary, by following the path of the target using a statistically-based target tracking algorithm with a Kalman filter. Multiple edge features, including position and intensity information, can be tracked simultaneously, permitting a high level of discrimination in the tracing process without invoking complicated heuristics.

In [17], a line-by-line image scan was used and intensity nonuniformity was the problem of interest. In the present study, a new algorithm is described and all three of the main segmentation problems are considered. Edges are traced beginning at an operator-defined starting point to simplify the data processing and focus on user-selected object boundaries. Intensity features obtained from regions on both sides of an edge can be combined with spatial edge coordinates in the tracking process. Global information in the form of pixel classification data and prior knowledge in the form of model parameter selection are used to influence the track direction. Spatial edge coordinates are interpolated to subpixel resolution for smooth contours while sharp transitions are also permitted by modeling target acceleration. This new algorithm is referred to as DTC, Dynamic Edge Tracing with classification. The objective of this study is to investigate its suitability as a tool for the segmentation of magnetic resonance (MR) head images.

The dynamic edge tracing algorithms represent the first application of target-tracking-based edge tracing to the segmentation of MR, head images. Early work on the use of target tracking for the purpose of edge tracing can be found in [18], although the algorithm described therein does not permit the formation of closed contours. Closed contours are formed in [16] but the object boundaries must be approximately convex such that line of sight to a central seed pixel is retained for all edge points. This may be

suitable, as intended, for certain tasks in ultrasound image analysis but the boundaries to be extracted from MR, head images are generally not convex. In DTC, nonconvex contours are readily accommodated.

The DTC algorithm will not always find closed contours and thus, must be considered as a work in progress. However, it is shown that when closed contours are formed, DTC can produce segmentation results comparable to methods based on clustering and on level sets. An example is also shown to illustrate the point that the information extracted by edge tracing is not encapsulated in either of the other two methods, and an example of how DTC can be used to add edge information to pixel classifications obtained from the FMRIB Automated Segmentation Tool (FAST). These results permit the statement that automatic edge tracing has value for medical image segmentation and also suggest that further research could improve the level of automation and the level of integration with other methods.

In the following, section 5.2 details the methods in four subsections. The first three describe the algorithms used in this investigation and the fourth gives the methods used for evaluation. Results from the application of these segmentation methods to MR, head images that include noise, intensity nonuniformity, and partial volume averaging are presented in section 5.3. Sections 5.4 and 5.5 provide further discussion and conclusions.

5.2 Methodology

5.2.1 Snake Automated Partitioning (SNAP)

The SNAP software application [19] was developed using level set algorithms in the Insight Segmentation and Registration Toolkit (ITK) [20] and is freely distributed as

open-source software. Version 1.0 has been used in this study. The SNAP tool implements two different types of level set algorithms, geodesic active contours [21] and a version of the region competition snakes algorithm based on [5] with minor modifications. The region competition method was used in this study since it was found to be less likely to overflow weak edges.

The SNAP tool provides three methods for implementing the level set curve evolution in discrete time and space: dense, narrow band, and sparse field. These are further described in [22]. The narrow band method, in which the level set function is updated at pixels near to the vicinity of the zero level set, has been used in this study. This is computationally more efficient than the dense method and the implementation permits operation on a single slice, if desired.

5.2.2 FMRIB Automated Segmentation Tool (FAST)

The FAST algorithm uses statistical clustering and a hidden Markov random field to perform unsupervised segmentation [6]. It was developed by the Oxford Centre for Functional Magnetic Resonance Imaging of the Brain (FMRIB) at Oxford University, U.K., and is part of the FMRIB Software Library (FSL) [23].

To perform a segmentation operation, the brain extraction tool (BET) [24] in the library is first applied to the 3D, MR image to isolate the brain region from the surrounding tissue regions. Subsequently, the FAST algorithm is applied to classify the pixels in the brain region into three classes, cerebrospinal fluid (CSF), white matter (WM), and gray matter (GM). The current versions of BET (1.2) and FAST (3.5) were used.

The FAST algorithm produces a hard segmentation, where after clustering each image pixel is assigned to the class of highest probability, as well as a normalized partial volume effect (PVE) estimate, which, for each pixel, consists of a value between zero and unity for each of the three classes. The hard segmentation was used in this evaluation since it produced the higher scores.

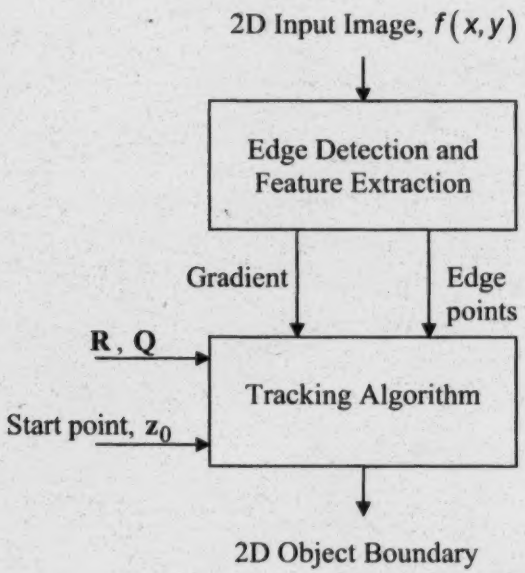

Figure 5-1. Processing Steps.

5.2.3 Dynamic Edge Tracing (DTC)

Figure 5-1 shows a block diagram of the major processing steps. Edge detection and edge feature extraction are performed on the input image and the results are used to drive a target tracking algorithm. Beginning at an operator-defined starting location, the tracking algorithm follows the path of a hypothetical target along an edge until the

starting point is revisited (a closed boundary is formed) or until no further edge data can be assigned to the track. In the event that the track terminates before revisiting the starting point, a second track is automatically propagated beginning at the starting location and progressing in a direction opposite to that of the first.

5.2.3.1 Edge Detection

Spatial edge coordinates are identified as zero crossings in the second derivative of a Gaussian filtered image [25],

$$f_{\sigma_c}(x,y) = G_{\sigma_c} * f(x,y) \tag{5-1}$$

$$\frac{\partial^2}{\partial n_j^2} f_{\sigma_c}(x,y) = 0 \tag{5-2}$$

where $f(x,y)$ is the image data, $f_{\sigma_c}(x,y)$ is the Gaussian filtered image, G_{σ_c} is a 2D, Gaussian filter with characteristic width, σ_c, and $\boldsymbol{n}_j$ is a unit vector in the operator direction. Directional operators are used and the operator direction takes on six values with equal, 30-degree intervals. Zero crossings are linearly interpolated in the operator direction to obtain subpixel resolution. The final set of spatial edge coordinates is the union of those identified in all operator directions and includes noise edges, analogous to clutter during the target tracking phase. In most approaches to edge detection, gradient thresholds are used to attempt separation of noise edges from true edges; however, no gradient threshold is used here. This avoids one of the difficulties with edge detection where the threshold may eliminate valid edge points in the image at places of relatively lower gradient, producing gaps in the desired boundary.

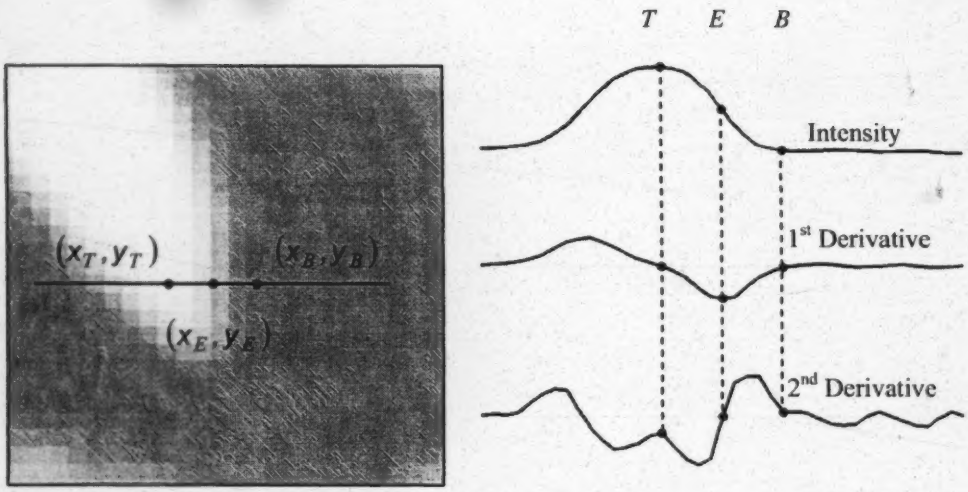

Figure 5-2. Edge Features.
The image fragment at left contains a line segment in the direction of the edge detection operator. Intensity, 1st and 2nd derivative waveforms for the line segment are plotted on the right. Three main points are identified. Edge coordinates (x_E, y_E) correspond to the zero crossing in the 2nd derivative. Point (x_T, y_T) is the location of the top of the intensity slope with intensity value, T. Point (x_B, y_B) is the location of the bottom of the intensity slope with intensity, B.

Features of interest are also extracted during the edge detection process, as shown in figure 5-2. For edge coordinates (x_E, y_E), a line is extended on each side in the operator direction. Examination of the 1st and 2nd derivatives of the intensity along this line allows the locations of the top and bottom of the edge slope to be determined, corresponding to points on either side of the zero crossing at E where the 2nd derivative curve flattens out, or creates a notch, after peaking. Zero crossings in the 1st or 2nd derivative curves limit the distance from E for placement of these points. The intensity on each side of the edge can then be determined, $T = f(x_T, y_T)$, $B = f(x_B, y_B)$. This information is combined

with the subpixel spatial coordinates to form an *edge point*, a point in a four-dimensional space,

$$\mathbf{z} = \begin{bmatrix} x_E & y_E & T & B \end{bmatrix}^T. \tag{5-3}$$

5.2.3.2 Target Tracking

To begin, it is assumed that a hypothetical, maneuvering target has followed the boundary of an object of interest in $f(x,y)$ and that measurements of its position have been captured as four-dimensional edge points, in the form of (5-3). The position and velocity, or state, of the target at discrete time steps is modeled by a linear relationship with white, additive process noise,

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{w}_k \tag{5-4}$$

where k is the step counter, $\mathbf{x}_k$ is the state vector at step k, $\mathbf{A}$ is the state transition matrix, and $\mathbf{w}_k$ is the process noise vector. Measurements of the target position, $\mathbf{z}_k$, at each step are also assumed to contain noise,

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \tag{5-5}$$

where $\mathbf{H}$ is the measurement matrix, and $\mathbf{v}_k$ is the measurement noise vector. Elements of $\mathbf{w}_k$ and $\mathbf{v}_k$ are assumed to have zero-mean, Gaussian distributions and the elements of $\mathbf{v}_k$ are assumed to be independent. Measurement $\mathbf{z}_k = \begin{bmatrix} x_{Ek} & y_{Ek} & T_k & B_k \end{bmatrix}^T$ is an edge point (5-3).

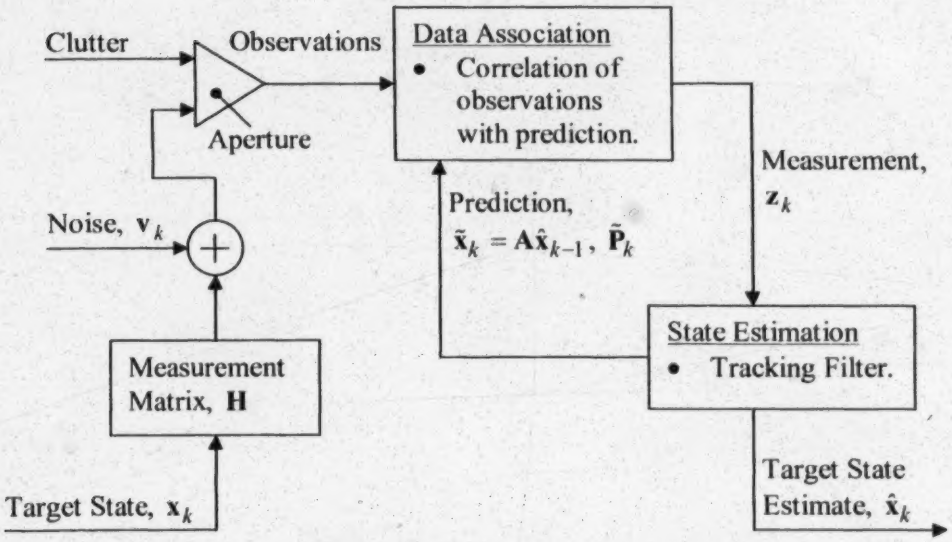

Figure 5-3. Tracking System Block Diagram

A target tracking algorithm is applied to find an estimate, $\hat{\mathbf{x}}_k$, of the target state at each step, k, along the edge. Finding these estimates will allow the path of the target to be drawn, an operation which is equivalent to that of finding the boundary of an object in the image. After tracking, the spatial dimensions of $\hat{\mathbf{x}}_k$, $k = 1..n$ for a track consisting of n edge points, are used to form the object boundary.

Figure 5-3 shows the block diagram of the target tracking system. A noisy measurement of target position along with nontarget clutter produce a population of candidate points, or observations, at each time step. The tracking system uses a *data association* process to select $\mathbf{z}_k$ from these observations. A *state estimation* process is then used to estimate the target state using the new measurement, and to form a state prediction which will be used by the data association process at the next step. This is

repeated for each step, k. Detailed descriptions of target tracking algorithms can be found in [26].

State estimation is performed using the Kalman filter [26], [27] followed by back-filtering using the Rauch-Tung-Striebel smoother [28]. The Kalman filter is the recursive solution to the discrete-time, linear, minimum mean square error estimation problem and the statistical estimator commonly used in dynamic tracking. It is defined with the assumptions of a linear, dynamic system and zero-mean, Gaussian noise. The well known Kalman filter equations for state estimation (5-6), (5-7), (5-8) and state prediction (5-9), (5-10) are

$$\mathbf{K}_k = \tilde{\mathbf{P}}_k \mathbf{H}^T \left(\mathbf{H} \tilde{\mathbf{P}}_k \mathbf{H}^T + \mathbf{R} \right)^{-1} \tag{5-6}$$

$$\hat{\mathbf{x}}_k = \tilde{\mathbf{x}}_k + \mathbf{K}_k \left(\mathbf{z}_k - \mathbf{H} \tilde{\mathbf{x}}_k \right) \tag{5-7}$$

$$\mathbf{P}_k = \left(\mathbf{I} - \mathbf{K}_k \mathbf{H} \right) \tilde{\mathbf{P}}_k \tag{5-8}$$

$$\tilde{\mathbf{x}}_{k+1} = \mathbf{A} \hat{\mathbf{x}}_k \tag{5-9}$$

$$\tilde{\mathbf{P}}_{k+1} = \mathbf{A} \mathbf{P}_k \mathbf{A}^T + \mathbf{Q} \tag{5-10}$$

where $\mathbf{K}_k$ is the Kalman filter gain, $\mathbf{P}_k$ is the a posteriori error covariance matrix, $\tilde{\mathbf{P}}_k$ is the a priori, or prediction error covariance, $\mathbf{R} = E\{\mathbf{v}_k \mathbf{v}_k^T\}$ is the measurement noise covariance matrix, $\tilde{\mathbf{x}}_k$ is the state prediction for step k, $\mathbf{I}$ is the identity matrix, $\mathbf{Q} = E\{\mathbf{w}_k \mathbf{w}_k^T\}$ is the process noise covariance matrix, and $E\{\cdot\}$ represents mathematical expectation. The innovation sequence is defined as $\mathbf{y}_k = \mathbf{z}_k - \mathbf{H} \tilde{\mathbf{x}}_k$ with covariance matrix, $\mathbf{S}_k = \mathbf{H} \tilde{\mathbf{P}}_k \mathbf{H}^T + \mathbf{R}$.

In the case of DTC, $\hat{\mathbf{x}}_k$, consists of components for position and velocity in four dimensions, $\hat{\mathbf{x}}_k = \left[\mathbf{p}_k^T \quad \Delta\mathbf{p}_k^T\right]^T$, where $\mathbf{p}_k$ is the estimate of position and $\Delta\mathbf{p}_k$ is the estimate of velocity. The position estimate, $\mathbf{p}_k$, exists in the same space as $\mathbf{z}_k$.

For edge tracing there is no explicit time dimension so the time, Δ, between successive tracking steps, k, is defined to be unity ($\Delta = 1$). With this definition, position change between successive tracking steps is equivalent to target velocity. Unequal changes in position from one step to another represent target acceleration.

The motion of the hypothetical target is modeled with a *target dynamic model* that assumes white noise acceleration with otherwise constant velocity. The acceleration is modeled as zero-mean, Gaussian noise with variance, σ_m^2. The one-dimensional, two-state (position and velocity) case for this model in discrete time is [26],

$$\mathbf{Q} = q\begin{bmatrix} \frac{\Delta^3}{3} & \frac{\Delta^2}{2} \\ \frac{\Delta^2}{2} & \Delta \end{bmatrix} \tag{5-11}$$

where q is the dynamics parameter and $q \propto \sigma_m^2$. Equation (5-11) can be directly extended from its one-dimensional representation to the four-dimensional case under consideration, producing a dynamics parameter for each dimension. A common parameter, $q = q_x = q_y$, is used for the two spatial dimensions and q_T, q_B are used for the respective intensity dimensions. These parameters describe the likelihood of a target maneuver in the corresponding dimension and can be used to influence track direction.

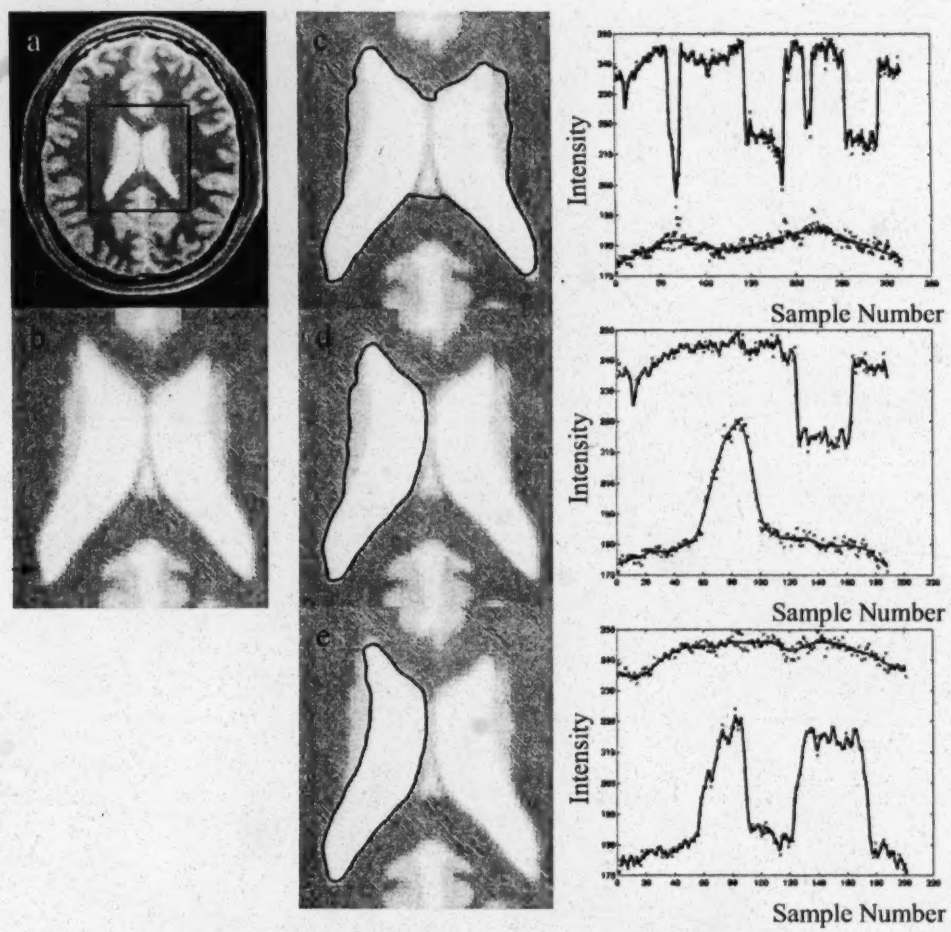

Figure 5-4. Intensity Dynamics Example.

Three different contours from the same starting point. (a) MNI PD slice 95 with 1% noise and 20% nonuniformity and region of interest identified; (b) region of interest with starting point identified (black arrow); (c) contour formed with $q_T = 50$, $q_B = 0.0001$; (d) contour formed with $q_T = 50$, $q_B = 0.05$; (e) contour formed with $q_T = 0.01$, $q_B = 50$. Track intensity variables are shown to the right of each with the position estimate in each dimension shown as a solid line and measurements marked as individual points. Tracks proceed from the starting point in a counterclockwise direction.

Figure 5-4 shows an example where selection of parameters, q_T and q_B, permits three different object boundaries to be found beginning from a single starting point. Selecting a low value for either q_T or q_B causes the tracking system to avoid transitions in the corresponding dimension whereas high values indicate that sharp transitions in the corresponding dimension are likely.

The transition from one state to another is modeled by simple kinematics and measurement is of four-dimensional target position only. Matrices, $\mathbf{A}$ and $\mathbf{H}$, therefore remain constant for all k,

$$\mathbf{A} = \begin{bmatrix} \mathbf{I} & \mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \tag{5-12}$$

$$\mathbf{H} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} \tag{5-13}$$

where $\mathbf{0}$ is a 4×4 zero matrix. The structure of $\mathbf{H}$ indicates position-only measurement.

Initial values, $\hat{\mathbf{x}}_1$ and $\mathbf{P}_1$, are determined using two edge points, $\mathbf{z}_0$ and $\mathbf{z}_1$. Measurement, $\mathbf{z}_0$, is the edge point at the operator-defined starting point and $\mathbf{z}_1$ is the first automatically-selected point. Following from [29],

$$\mathbf{P}_1 = \begin{bmatrix} \mathbf{R} & \mathbf{R} \\ \mathbf{R} & 2\mathbf{R} \end{bmatrix}. \tag{5-14}$$

Estimate, $\hat{\mathbf{x}}_1$, is chosen so that initial target motion is restricted to the spatial dimensions,

$$\mathbf{p}_1 = \begin{bmatrix} x_{E1} \\ y_{E1} \\ (T_0 + T_1)/2 \\ (B_0 + B_1)/2 \end{bmatrix}, \tag{5-15a}$$

$$\Delta \mathbf{p}_1 = \begin{bmatrix} x_{E1} - x_{E0} \\ y_{E1} - y_{E0} \\ 0 \\ 0 \end{bmatrix}. \qquad \text{(5-15b)}$$

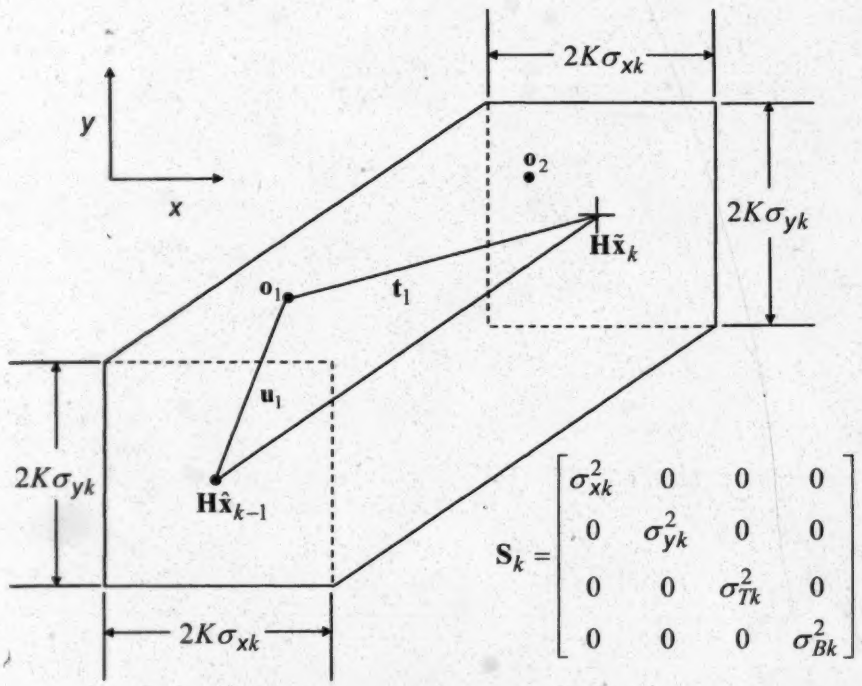

Figure 5-5. Data Association.

Association gate (outer, solid line, hexagonal shape) in the two spatial dimensions with predicted position, $\mathbf{H}\tilde{\mathbf{x}}_k$, previous position estimate, $\mathbf{H}\hat{\mathbf{x}}_{k-1}$. Two observations, $\mathbf{o}_1$ and $\mathbf{o}_2$ are shown with $\mathbf{t}_1$ and $\mathbf{u}_1$, distances to $\mathbf{o}_1$.

The *data association* process determines $\mathbf{z}_k$ using an association gate in the measurement space, evaluation of an objective function, and verification. The association gate in two spatial dimensions is shown in figure 5-5. The gate size characteristic in each dimension is determined by the standard deviation of the innovation, obtained from diagonal, covariance matrix, $\mathbf{S}_k$, and a constant multiplier, $2K$, with $K \cong 3$. In the spatial dimensions the gate includes the predicted position, $\mathbf{H}\tilde{\mathbf{x}}_k$, and the most recent position estimate, $\mathbf{H}\hat{\mathbf{x}}_{k-1}$. This facilitates the sequencing of valid edge points and recognition of sharp transitions. In the intensity dimensions, the gate is rectangular and centered on $\mathbf{H}\tilde{\mathbf{x}}_k$.

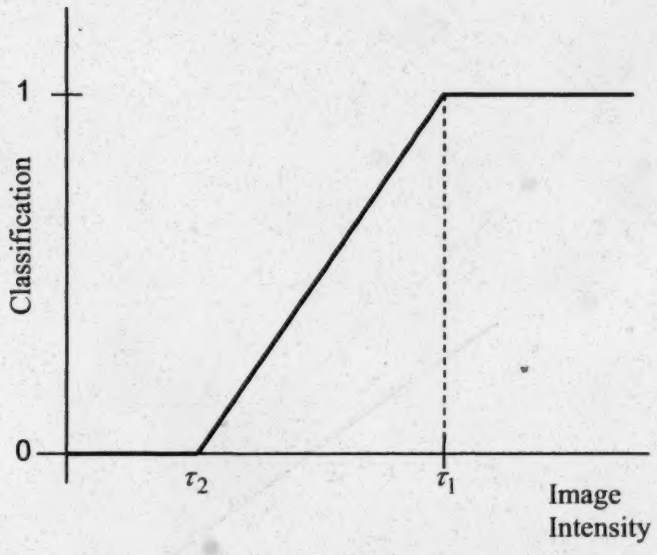

Figure 5-6. Threshold Classification.
Two thresholds, τ_1 and τ_2 are selected by the operator. Linear scaling is applied for intensity values between the two thresholds. The pixels in the classification image range from zero to unity.

Edge points within the gate that have not been previously included in the track are classed as observations. All observations, $\mathbf{o}_i$, at step k are examined according to the Mahalanobis distance measures (see also figure 5-5)

$$\breve{d}_i = \sqrt{\mathbf{u}_i^T \mathbf{G}_k^{-1} \mathbf{u}_i} \tag{5-16}$$

$$\hat{d}_i = \sqrt{\mathbf{t}_i^T \mathbf{S}_k^{-1} \mathbf{t}_i} \tag{5-17}$$

$$d_i = \hat{d}_i + \breve{d}_i \tag{5-18}$$

where $\mathbf{t}_i = \mathbf{o}_i - \mathbf{H}\tilde{\mathbf{x}}_k$, $\mathbf{u}_i = \mathbf{o}_i - \mathbf{H}\hat{\mathbf{x}}_{k-1}$, and $\mathbf{G}_k = \mathbf{H}\mathbf{P}_{k-1}\mathbf{H}^T + \mathbf{R}$. A normalized distance measure, D_i, is then computed as,

$$D_i = 1 - \left(d_i - dmn_k\right) / \left(dmx_k - dmn_k\right) \tag{5-19}$$

where dmn_k and dmx_k are the minimum and maximum limits of d_i. Distance dmx_k occurs at the extremity of the association gate and dmn_k occurs along the line between $\mathbf{H}\hat{\mathbf{x}}_{k-1}$ and $\mathbf{H}\tilde{\mathbf{x}}_k$.

A normalized, gradient strength measure, M_i, is computed as,

$$M_i = \left| g_i / gmx \right| \tag{5-20}$$

where g_i is the gradient strength of $\mathbf{o}_i$, and gmx is the peak gradient in the image.

Pixel classification is also utilized in the data association process. A normalized, classification image, $f_c(x,y)$, is formed by applying the mapping shown in figure 5-6 to $f_{\sigma_c}(x,y)$. Upper threshold, τ_1, and lower threshold, τ_2, form saturation limits and the intensity values between these thresholds are linearly scaled to within the range, zero to unity. A binary threshold is produced for the case where $\tau_1 = \tau_2$. The thresholds are

selected by the operator to produce high contrast between the object of interest and its surroundings.

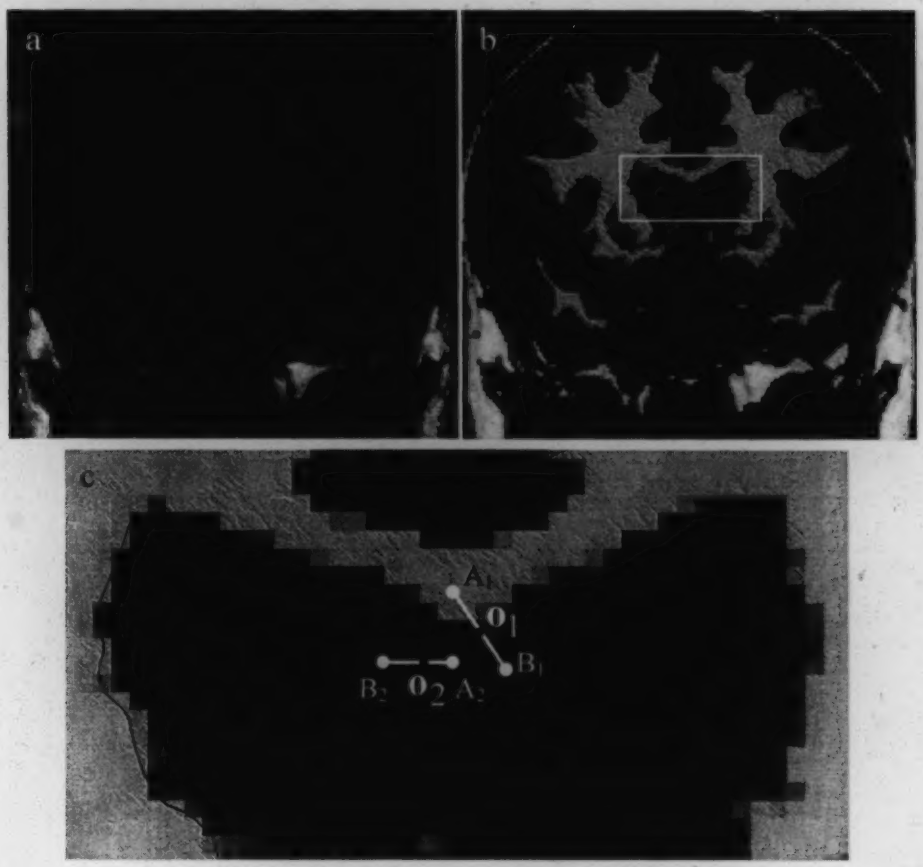

Figure 5-7. Use of the Classification Image.
(a) $f(x,y)$, slice 80 from IBSR_01; (b) classification image, $f_c(x,y)$, is shown as light-gray pixels overlaid on $f(x,y)$. White box identifies the zoom region; (c) Zoom region with partial track (black line). The observation with sample points A_1, and B_1 is preferred over the one with sample points A_2, and B_2 since segment A_1B_1 spans the classification boundary whereas A_2B_2 does not.

A classification measure, C_i, evaluates the extent to which $\mathbf{o}_i$ resides on the object boundary, the transition zones in $f_c(x,y)$,

$$C_i = CB_i \cdot CT_i \tag{5-21}$$

where $CT_i = f_c(x_{Ti}, y_{Ti})$ is the classification value sampled at the top of the edge slope for $\mathbf{o}_i$ and $CB_i = \left[1 - f_c(x_{Bi}, y_{Bi})\right]$ is the complement of the value at bottom of slope. An example is shown in figure 5-7. The image $f(x,y)$ is shown in panel (a) and $f_c(x,y)$, identified as light-gray pixels, is overlaid on $f(x,y)$ in panel (b). Panel (c) shows a zoomed region containing a partial track that is propagating toward the right. Observations, $\mathbf{o}_1$ and $\mathbf{o}_2$, are among those within the association gate. Sample points marked A_1 and B_1, corresponding to (x_T, y_T) and (x_B, y_B), respectively, for $\mathbf{o}_1$, are used to sample $f_c(x,y)$ and produce the classification measure, C_1. Similarly, points A_2 and B_2 are used to produce C_2. Observation, $\mathbf{o}_1$, is preferred over $\mathbf{o}_2$ (i.e. $C_1 > C_2$) since segment A_1B_1 spans the classification boundary whereas A_2B_2 does not. Thus, $f_c(x,y)$ guides the edge tracer, allowing the classification result to assist in forming the boundary. The edge tracer retains some freedom relative to the classification, though, as edge points are still used to form the boundary. This allows the boundary to deviate from the classification to a limited extent as is visible in figure 5-7c where the black-line boundary can encroach upon the classification or, alternatively, pull away from it. This can also be viewed as a mechanism for using edge information to refine the classification result.

The objective function, Ψ, is the product of these three normalized measures. For $\mathbf{o}_i$,

$$\Psi_i = D_i \cdot M_i \cdot C_i . \tag{5-22}$$

Observations are ranked according to this score which will be highest for those with relatively high gradient, lying in the current direction of travel, and which classify well as boundary points of the object of interest. When multiple observations have equal value, the observation with minimum $\breve{d}_i$, that is, producing lower velocity, is ranked higher.

Measurement, $\mathbf{z}_k$, is taken as the highest-ranked observation that passes verification of track direction and step size. There are two possible directions for a track to proceed along an edge and a reversal of direction is not desirable. The track direction is determined by translating the velocity vector (spatial dimensions) to (x_{Ek}, y_{Ek}) and examining the locations of (x_{Tk}, y_{Tk}) and (x_{Bk}, y_{Bk}) (one right and one left) relative to its orientation. The direction produced by $\mathbf{z}_k$ must be consistent with the existing track direction.

A gap of greater than one pixel width between edge points may be due to a minor intensity variation or may separate two different edges that have similar intensity features. Interpolation of intensity data in $f_{\sigma_c}(x,y)$ is performed from (x_{Tk-1}, y_{Tk-1}) to (x_{Tk}, y_{Tk}) and from (x_{Bk-1}, y_{Bk-1}) to (x_{Bk}, y_{Bk}). To be acceptable, accumulated intensity variation along either path must be less than the previous edge height. That is, for interpolated intensity values, I_j, $j = 1..N$, along either path, verification requires,

$$\left| \left(\sum_{j=2}^{N} \left| I_j - I_{j-1} \right| \right) - \left| I_1 - I_N \right| \right| < \left(f_{\sigma_c}(x_{Tk-1}, y_{Tk-1}) - f_{\sigma_c}(x_{Bk-1}, y_{Bk-1}) \right). \tag{5-23}$$

This step ensures, for example, that the track will not jump across a narrow projection of the object of interest when both sides of the projection fall within the association gate.

5.2.4 Evaluation

The Tanimoto similarity measure [30], TS, is defined as the ratio of the number of elements in the intersection of two sets to the number of elements in their union. For two sets, A, B,

$$TS = \eta(A \cap B)/\eta(A \cup B). \tag{5-24}$$

where η stands for the cardinality of a set. This is used to evaluate segmented regions, binary images consisting of all pixels within the boundary of interest.

The Hausdorff distance [31] is defined as the maximum of the closest-point distances between two contours. For contours, $\boldsymbol{A} = \{\boldsymbol{a}_1, \boldsymbol{a}_2, ..., \boldsymbol{a}_m\}$, $\boldsymbol{B} = \{\boldsymbol{b}_1, \boldsymbol{b}_2, ..., \boldsymbol{b}_n\}$, where each element in each contour is an ordered pair of spatial coordinates, (x, y), the closest-point distance for point $\boldsymbol{a}_i$ to contour $\boldsymbol{B}$ is,

$$h(\boldsymbol{a}_i, \boldsymbol{B}) = \min_j \left\| \boldsymbol{b}_j - \boldsymbol{a}_i \right\| \tag{5-25}$$

and the Hausdorff distance for the two contours is,

$$HD(\boldsymbol{A}, \boldsymbol{B}) = \max\left(\max_i \left\{ h(\boldsymbol{a}_i, \boldsymbol{B}) \right\}, \max_j \left\{ h(\boldsymbol{b}_j, \boldsymbol{A}) \right\} \right). \tag{5-26}$$

This is used to evaluate boundary shape and supplements TS by detecting contour discrepancies in the form of narrow projections where the error area is small. Good contour similarity is indicated when HD is low and TS is near unity.

5.3 Results

Synthetic and real MR images were used to examine the DTC, FAST, and SNAP segmentation algorithms under conditions of noise, intensity nonuniformity, and partial volume averaging. Brain WM regions were selected for segmentation because of their relatively large size and their detailed, convoluted, and often low-contrast boundaries. Algorithm performance in noise and intensity nonuniformity was evaluated using synthetic images from the Montreal Neurological Institute (MNI) database, normal brain, 1 mm spacing, proton density (PD), and scaled, raw-byte data format [32]. These data contain some partial volume effects but further partial volume tests were performed in a corpus callosum study using real MR images from the Internet Brain Segmentation Repository (IBSR) V2.0 database [33]. Known and manual segmentations, respectively, are available as part of these databases and were used to form a reference region, the expected segmentation, and a reference contour, the outer object boundary.

The parameters for each segmentation method were adjusted to achieve maximum *TS* when compared with the reference region. This was done to have a point of comparison for the three relatively diverse segmentation algorithms. The parameter selection was performed by an experienced operator with full knowledge of the expected segmentation, allowing an optimal, or nearly optimal, result to be determined in each case. The *HD* between the reference contour and the outer contour in each segmentation result was also computed.

The resulting segmentations were validated visually and accepted only if they were found to be representative of the underlying anatomy. Candidate segmentations that included large portions of neighbouring tissue regions, for example, were excluded.

5.3.1 Parameter Settings

For DTC, parameters K, q, q_T, q_B, $\mathbf{R}$, σ_c, τ_1, τ_2 affect system operation and of these, $\mathbf{R}$, K, and q were preselected. Matrix $\mathbf{R}$ is a 4×4 diagonal matrix selected to have elements $\sigma_x^2 = \sigma_y^2 = 0.2$, suggesting typical spatial edge placement errors of less than 0.5 pixels with worst case approximately 2 pixels, and $\sigma_T^2 = \sigma_B^2$ defined by an estimate of the image noise variance, σ_N^2. Image noise variance estimation was performed using the *Average* method, highest ranked in the evaluation of [34], and rounded to the nearest 5 squared gray levels. A value of $\sigma_N^2 = 3$ was used for synthetic images without noise. Also, $K = 3.3$ was selected to set a gate size covering 99.9% of the related Gaussian distributions, and $q = 0.2$ was found experimentally to be useful for contours with significant spatial variation. Parameters σ_c, q_T, q_B, τ_1, τ_2 remained to be selected by the operator. Parameters σ_c, τ_1, τ_2 were set once for each 2D image, leaving two parameters, q_T and q_B, to be selected for each track. Multiple parameter sets sometimes produced similar results and there were conditions where only one, or neither, intensity dimension was needed in the tracking operation.

Since the WM region to be extracted included smaller, internal or external, subregions, several contours were used to complete the segmentation; however, only one contour was allowed for each region or subregion and only one starting point was permitted for each contour. Track propagation was halted if one track encountered another. In cases where a closed contour was not formed, the contour was accepted only if a valid segmentation could be produced by linking the two end points with a small straight line.

For SNAP, the operator initially defined a 3D, rectangular region bounding the segmentation. For images where the pixels were not cubes, such as with the IBSR data, the region was resampled to form cuboid pixels. Three preprocessing parameters, an upper threshold, ϕ_1, a lower threshold, ϕ_2, and a smoothness parameter, ς, were required to form an operator-defined probability distribution over pixel intensities for the desired object. Two snake parameters were also required, a balloon force, α, and a curvature force, β. All of these parameters were set on a per image basis. It was found, though, that a balloon force of $\alpha = 1.0$ was suitable for all of the images. The SNAP application also contains an option for gray level scaling, although this was not used.

Typically, four seed bubbles were used to initialize the SNAP segmentation and these were placed in regions corresponding to high WM content. The number of seed bubbles was determined arbitrarily and was made greater than unity simply to reduce the processing time. The level set algorithm was permitted to find the extremities of the WM without being influenced by targeted seed placement on the part of the operator.

Segmentation in FSL using BET and FAST is fully automatic. It is possible to set threshold parameters to affect the point where BET defines the brain boundary but only the default parameters were used in this study. It is also possible to post-process the segmentation, for example using morphological operations to remove obvious misclassifications visible as small, off-brain outliers but no post-processing steps were taken. These choices preserve the fully-automatic nature of this algorithm.

5.3.2 Noise and Intensity Nonuniformity

Noise and intensity nonuniformity were examined using slice 95 of the MNI PD synthetic image dataset. Results from DTC, FAST, and SNAP, were compared with

expected results after segmentation of cerebral WM. The synthetic image database contains a fuzzy classification for each image pixel. Crisp WM pixels, those for which WM is the class of highest membership, were used in the comparison. No penalty was assessed for inclusion or exclusion of pixels exhibiting minor WM content. The outer boundary of the crisp WM region was used as the reference contour for *HD* calculations.

The FAST result was determined by processing the entire 3D image whereas SNAP and DTC only processed slice 95. For SNAP, this arrangement was selected to reduce processing time.

Figure 5-8a-c shows *TS* versus noise level for each method at 0%, 20%, and 40% nonuniformity and table 5-1 gives the *HD* values. The FAST result shows a dip in *TS* at low noise levels in each of the three cases. This was found to be due to misclassification along the GM-WM interface, with a nonuniform distribution of errors.

Since DTC can be guided by pixel classification other than that formed by τ_1 and τ_2, a test was run with

$$f_C(x,y) = 1 - f_{F_{WM}}(x,y)$$

where $f_{F_{WM}}(x,y)$ is the FAST WM segmentation. The complement operation employed simply matches high classification value with high intensity, as required in DTC. Table 5-2 contains the results for each of the three cases. In each case, the DTC-FAST combination improves on the result from FAST alone, and in two of the three, the *HD* of the combination is better than that produced by either method.

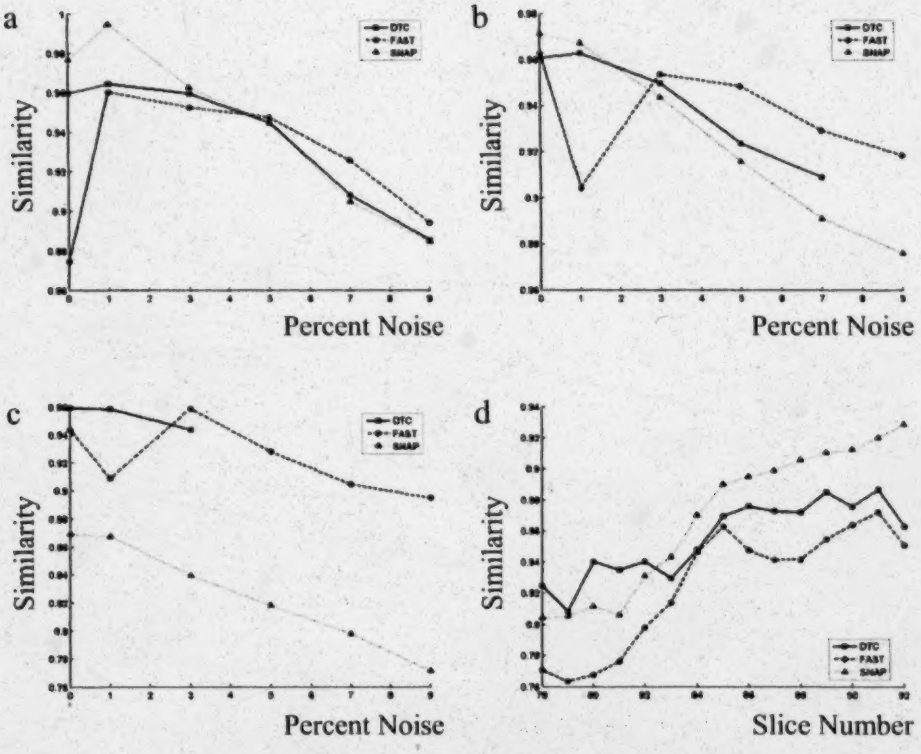

Figure 5-8. Similarity Measure.

Similarity measure results (a)-(c) vs. noise level for MNI PD Slice 95 (a) 0% nonuniformity; (b) 20% nonuniformity; (c) 40% nonuniformity; (d) dataset IBSR_01 slices 78 to 92.

Table 5-1 – Hausdorff Distance (pixels) for MNI Slice 95.

	Percent Noise						
	0	1	3	5	7	9	
	0% Nonuniformity						
σ_N^2	0	5	30	70	105	175	avg
DTC	8.1	6.1	7.1	8.2	10.1	10.1	8.3
FAST	10.0	5.0	6.0	4.1	10.0	10.0	7.5
SNAP	7.2	5.0	7.1	5.8	8.2	8.6	7.0
	20% Nonuniformity						
σ_N^2	0	5	25	65	105	145	avg
DTC	6.1	7.4	8.0	11.1	9.7	-	8.5
FAST	5.0	10.0	5.8	9.2	9.2	8.9	8.0
SNAP	5.8	5.8	8.5	10.3	8.9	12.1	8.6
	40% Nonuniformity						
σ_N^2	0	5	20	55	95	135	avg
DTC	5.8	6.2	9.5	-	-	-	7.2
FAST	11.4	10.0	5.0	10.4	17.3	10.8	10.8
SNAP	13.9	16.6	19.2	19.2	20.2	20.6	18.3

Table 5-2 – Metric Comparison for DTC-FAST Combination.

	0% Noise, 0% NonU		1% Noise, 20% NonU		1% Noise, 40% NonU	
	TS	HD	TS	HD	TS	HD
DTC	0.96	8.1	0.96	7.4	0.96	6.2
FAST	0.87	10.0	0.90	10.0	0.91	10.0
DTC-FAST	0.95	8.5	0.96	6.0	0.96	5.9

Table 5-3 – Hausdorff Distance (pixels) - IBSR_01 Slices.

	Slice Number							
	78	79	80	81	82	83	84	85
DTC	8.3	8.5	4.0	12.1	11.5	9.3	5.3	2.7
FAST	6.7	6.7	14.9	12.1	5.7	4.2	4.1	2.2
SNAP	7.2	7.3	10.2	4.0	2.2	2.0	1.4	2.0
	Slice Number							
	86	87	88	89	90	91	92	avg
DTC	2.3	2.9	6.5	3.3	3.5	6.7	5.5	6.2
FAST	5.4	3.6	6.7	7.2	7.1	5.7	6.0	6.6
SNAP	1.4	1.4	6.3	2.8	3.2	2.2	7.3	4.1

5.3.3 Partial Volume Averaging

Partial volume averaging was further examined using real images and manual segmentations from dataset IBSR_01 [33], a set of consecutive 2D, coronal, T1-weighted, MR images with 1.5 mm slice thickness. These images have low noise and have been preprocessed by an intensity correction algorithm so that partial volume averaging is the main segmentation difficulty.

The test included fifteen consecutive images (slices 78 to 92) containing the genu of the corpus callosum, a WM region. The higher numbered slices are the most anterior and the simplest. In slices 91 and 92 the WM region is largely uninterrupted whereas from slice 90 and proceeding to lower numbers, the WM region increasingly surrounds other structures, first the lateral ventricles and then relatively low-contrast basal nuclei. The FAST method used the entire 3D image as input, SNAP was provided with a bounding box encompassing the desired WM region, and DTC operated on each 2D image individually. Figure 5-8d shows *TS* versus slice number and table 5-3 gives *HD* values.

Figure 5-9 shows images for slice 80. Significant partial volume effect occurs at the extremities of the WM region and all methods were affected by it. The DTC algorithm appears to be better able to avoid the GM, basal nuclei which both of the other methods classify largely as WM.

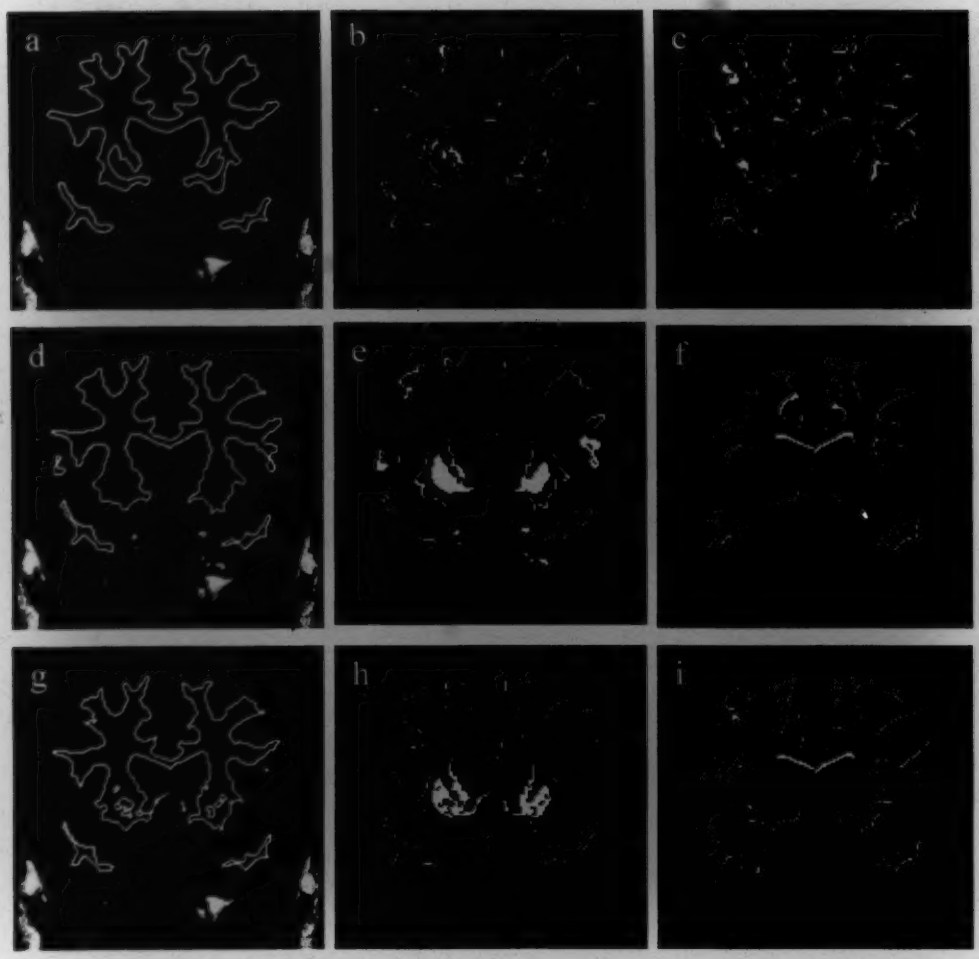

Figure 5-9. IBSR_01 Slice 80.

Contours, false positives and false negatives (respectively). For the false positives and false negatives, gray represents manual segmentation and error pixels are shown in white. (a), (b), (c) for DTC; (d), (e), (f) FAST; (g), (h), (i) SNAP.

5.3.4 Execution Time

All tests were performed on a 3.2 GHz PC. The DTC algorithm runs under Matlab® [35] whereas the SNAP and FAST programs are compiled software executables.

For DTC, execution time was approximately 5 milliseconds per contour point. The average outer contour in the MNI images, for example, contained 2056 points.

The SNAP algorithm operated at approximately 12.5 iterations per second with the MNI images. The number of iterations required to reach a stable solution increased with the level of difficulty of the image. Between 600 and 1200 iterations were typically required.

The BET and FAST algorithms operated on full 3D images in all cases. The brain region for each was extracted by BET in less than 10 seconds and the FAST segmentation required approximately 600 seconds for each of the MNI image volumes. Each MNI image volume contains 181x217x181 pixels.

5.4 Discussion

The results demonstrate that there is no one algorithm that is uniformly superior to the others. The SNAP method is best at low levels of nonuniformity but its performance degrades as nonuniformity is increased. This is reasonable to expect since the object intensity distribution is defined using globally applied thresholds. The FAST algorithm is best at the higher levels of noise but unusually high numbers of pixel misclassifications occur at the GM-WM boundary in some cases. The DTC algorithm is competitive with the others in all cases where a closed boundary could be formed. However, the combination of high noise and high intensity nonuniformity appears to be a greater

problem for DTC than for either FAST or SNAP. This is shown in figure 5-8a-c where DTC is unable to form a closed contour in four of the eighteen synthetic image test cases, particularly toward the limits of the noise and nonuniformity ranges.

As shown in figure 5-9, both FAST and SNAP misclassify the GM of the basal nuclei as WM even though a visible boundary exists. By comparison, the DTC method is able to follow the boundary, producing fewer errors. This example shows that the edge tracing result is not a subset of that produced by the other methods and this fact indicates the importance of continued research into recursive edge following methods.

In addition, DTC can combine edge information with pixel classification results from other methods. An example of this is shown in table 5-2 where the FAST WM classification was used to guide DTC as a replacement for the operator-defined threshold classification, $f_c(x,y)$. This produced an improvement over the FAST result in cases of high misclassification at the GM-WM interface as the edge tracer was able to refine the FAST result. The benefit of this approach, for DTC, is that the operator is relieved of the task of choosing appropriate thresholds to form $f_c(x,y)$, although it is an advantage only when the misclassifications occur along the boundary. As an extension, one could imagine the use of an inference engine in the data association process to permit additional information, including knowledge, to be used to guide the edge tracer.

The contours generated by DTC are very smooth due to the subpixel-resolution edge points and the statistical position estimate. The ability to form sharp corners is also retained, as is visible in figure 5-9. Also, the DTC algorithm can be easily expanded or adjusted. Properties of the edge that can be described by Gaussian distributions may be

tracked explicitly while other parameters can be included by modifying the objective function in the data association process.

Some problems remain, however, and further development will be required to solve them. Self intersection has not been addressed in the DTC algorithm although careful parameter selection often will permit a nonintersecting contour to be formed, as in the results presented here. Also, starting point and direction of travel can affect the resulting contour, especially in cases of partial volume averaging. Contour formation is sensitive to the sequence of edge points because the next-point selection in the data association process currently uses a nearest-neighbour criterion. Evaluation of multiple paths may produce better contours.

The parameters q, q_T, and q_B were determined experimentally. In a typical target tracking system, variables of this type would be determined by a tuning process, often involving simulation of anticipated tracking scenarios. In a similar manner, further studies could be done to develop a set of typical parameter settings for a range of tissue interfaces and image classes. Alternatively, it may be possible to automatically determine some of these parameters using an adaptive filter [36].

Although only a single set of parameters was used for each contour in these tests, it is clear that there are situations where a single set is not sufficient to form a closed boundary. In these scenarios, segmentation may be performed by a procedure in which the operator initiates several automatic tracks and then manually prunes and links the tracks together to form a closed boundary. Semiautomatic operation like this should still be valuable in relieving operator tedium when compared with manual tracing.

Finally, a note concerning the use of Gaussian noise models. In cases where the process noise or the measurement noise in the spatial dimensions of $\mathbf{z}_k$ is not Gaussian, the Kalman filter is still the best linear, minimum mean square error estimator. For the intensity dimensions, intensity noise is often modeled as Gaussian in MR images although the distribution is more accurately described as Rician. This is, however, approximately Gaussian for signal to noise ratio (SNR) greater than 3 [37], where $SNR = A/\sigma_N$ is the ratio of the signal amplitude in the absence of noise to the noise standard deviation. In the image background, where there is no MR signal, the noise follows a Rayleigh distribution. MR images from clinical scanners very often exhibit SNR well above 3 so the assumption of Gaussian noise distributions in these images is often valid.

5.5 Conclusion

An advanced edge tracing algorithm based on dynamic target tracking is capable of identifying object boundaries in complex, MR, medical images. The tracking algorithm operates in multiple dimensions, permitting the seamless use of multiple features to help discriminate object boundaries. Candidate edge points are ranked by evaluation of an objective function wherein local and global image information are combined. The algorithm is able to utilize pixel classification results from other methods and improve upon them in cases where misclassifications exist along the boundary of interest.

Comparison with results from FAST and SNAP demonstrates that the edge tracing algorithm can produce comparable results in images with realistic levels of noise, intensity nonuniformity and partial volume averaging. Furthermore, object information

extracted by edge tracing is not a subset of that extracted by the other methods, as was shown along low-contrast boundaries. These are important results for edge tracing algorithms, in general. Further research may permit improvements in the level of automation and in the level of integration with other methods.

5.6 Acknowledgment

The synthetic MR data sets were provided by the McConnell Brain Imaging Centre of the Montreal Neurological Institute, McGill University and are available at http://www.bic.mni.mcgill.ca/brainweb/.

The real MR brain data sets and their manual segmentations were provided by the Center for Morphometric Analysis at Massachusetts General Hospital and are available at http://www.cma.mgh.harvard.edu/ibsr/.

5.7 References

[1] L.P. Clarke, R.P. Velthuizen, M.A. Camacho, J.J. Heine, M. Vaidyanathan, L.O. Hall, R.W. Thatcher, M.L. Silbiger, "MRI segmentation: Methods and applications," *Magnetic Resonance Imaging*, Vol. 13, No. 3, 1995, pp. 343-368.

[2] K.S. Fu and J.K. Mui, "A survey on image segmentation," *Pattern Recognition*, Vol. 13, 1981, pp. 3-16.

[3] D. Pham, C. Xu, J. Prince, "Current methods in medical image segmentation," *Annual Review of Biomedical Engineering*, Vol. 2, 2000, pp. 315-337.

[4] T. McInerney, D. Terzopoulos, "Deformable models in medical image analysis: A survey," *Medical Image Analysis*, Vol. 1, No. 2, 1996, pp. 91-108.

[5] S.C. Zhu, A. Yuille, "Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18, No. 9, 1996, pp. 884-900.

[6] Y. Zhang, M. Brady, S. Smith, "Segmentation of brain MR images through a hidden Markov random field model and the expectation maximization algorithm," *IEEE Transactions on Medical Imaging*, Vol. 20, No. 1, 2001, pp. 45-57.

[7] I. Pitas, "Digital Image Processing Algorithms," Prentice-Hall, 1993.

[8] A.X. Falcão, J.K. Udupa, F.K. Miyazawa, "An ultra-fast user-steered image segmentation paradigm: Live wire on the fly," *IEEE Transactions on Medical Imaging*, Vol. 19, No. 1, 2000, pp. 55-62.

[9] M. Lineberry, "Image segmentation by edge tracing," in: Proc. of SPIE, The International Society for Optical Engineering, Vol. 359, *Applications of Digital Image Processing IV*, San Diego, Calif., USA, 1982, pp. 361-368.

[10] V. Rakotomalala, L. Macaire, M. Valette, P. Labalette, Y. Mouton, J.G. Postaire, "Bidimensional retinal blood vessel reconstruction by a new color edge tracking procedure," in: IEEE Southwest Symposium on Image Analysis and Interpretation, April 1998, pp. 232-237.

[11] Y.B. Chen, O.T.-C. Chen, "Robust fully-automatic segmentation based on modified edge-following technique," in: Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing, 2003, pp. III_333-III_336.

[12] V. Barrios, J. Torres, G. Montilla, L. Hernandez, N. Rangel, A. Reigosa, "Cellular edge detection using a trained neural network explorer," in: Proc. 16th Int. Conf. IEEE Engineering in Medicine and Biology Society, Vol. 2, 1994, pp. 1075-1076.

[13] H. Iwata, T. Agui, H. Nagahashi, "Boundary detection of color images using neural networks," in: Proc. IEEE Int. Conf. on Neural Networks, Vol. 3, 1995, pp. 1426-1431.

[14] H. Soltanian-Zadeh, J.P. Windham, "A multiresolution approach for contour extraction from brain images," *Medical Physics*, Vol. 24, No. 12, 1997, pp. 1844-1853.

[15] S. Mahamud, L.R. Williams, K.K. Thornber, K. Xu, "Segmentation of multiple salient closed contours from real images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 4, 2003, pp. 433-444.

[16] P. Abolmaesumi, M.R. Sirouspour, "An interacting multiple model probabilistic data association filter for cavity boundary extraction from ultrasound images," *IEEE Transactions on Medical Imaging*, Vol. 23, No. 6, 2004, pp. 772-784.

[17] D.J. Withey, Z.J. Koles, W. Pedrycz, "Dynamic edge tracing for 2D image segmentation," in: Proc. 23rd Int. Conf. IEEE Engineering in Medicine and Biology Society, Vol. 3, Oct. 2001, pp. 2657-2660.

[18] M. Basseville, B. Espiau, J. Gasnier, "Edge detection using sequential methods for change in level – Part I: A sequential edge detection algorithm," *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. ASSP-29, No. 1, 1981, pp. 24-31.

[19] http://www.itksnap.org/

[20] http://www.itk.org

[21] V. Caselles, R. Kimmel, G. Sapiro, "Geodesic active contours," *International Journal of Computer Vision*, Vol. 22, 1997, pp. 61-79.

[22] R.T. Whitaker, "A level-set approach to 3D reconstruction from range data," *International Journal of Computer Vision*, Vol. 29, No. 3, 1998, pp. 203-231.

[23] S.M. Smith, M. Jenkinson, M.W. Woolrich, C.F. Beckmann, T.E.J. Behrens, H. Johansen-Berg, P.R. Bannister, M. De Luca, I. Drobnjak, D.E. Flitney, R.K. Niazy, J. Saunders, J. Vickers, Y. Zhang, N. De Stefano, J.M. Brady, P.M. Matthews, "Advances in functional and structural MR image analysis and implementation as FSL," *NeuroImage*, Vol. 23, 2004, pp. S208-S219.

[24] S.M. Smith, "Fast robust automated brain extraction," *Human Brain Mapping*, Vol. 17, No. 3, 2002, pp. 143-155.

[25] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8, No. 6, 1986, pp. 679-698.

[26] S. Blackman, R. Popoli, "Design and Analysis of Modern Tracking Systems," Artech House, 1999.

[27] R.E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME - Journal of Basic Engineering*, March, 1960, pp. 35-45.

[28] A. Gelb, ed. "Applied Optimal Estimation," MIT Press, Cambridge MA, 1974.

[29] Y. Bar-Shalom, T.E. Fortmann, "Tracking and Data Association," Academic Press, 1988.

[30] S. Theodoridis, K. Koutroumbas, "Pattern Recognition," Academic Press, 1999.

[31] D. Huttenlocher, G. Klanderman, W. Rucklidge, "Comparing images using the Hausdorff distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 9, 1993, pp. 850-863.

[32] http://www.bic.mni.mcgill.ca/brainweb/

[33] http://www.cma.mgh.harvard.edu/ibsr/

[34] S.I. Olsen, "Estimation of noise in images: An evaluation," *CVGIP: Graphical Models and Image Processing*, Vol. 55, No. 4, 1993, pp. 319-323.

[35] http://www.themathworks.com

[36] F. Wang and V. Balakrishnan, "Robust Kalman filters for linear time-varying systems with stochastic parametric uncertainties," *IEEE Transactions on Signal Processing*, Vol. 50, No. 4, 2002, pp. 803-813.

[37] H. Gudbjartsson, S. Patz, "The Rician distribution of noisy MRI data," *Magnetic Resonance in Medicine*, Vol. 34, 1995, pp. 910-914.

Chapter 6

Discussion and Conclusions

6.1 Progression of Development

Chapters 3 to 5, containing descriptions of the dynamic edge tracing concepts, also identify a progression of development. To facilitate a comparison, the method of chapter 3 will be referred to as DTA, the method of chapter 4 as DTB, and the methods of chapter 5 by the symbol used there, DTC.

In chapter 3, DTA utilizes a line-by-line scan of the image to simulate the time dimension as well as an algorithm for tracking multiple targets where new tracks are started for any edge pixels that cannot be associated with an existing track as the line-by-line scan progresses. Although developed independently, this is similar to the method of Basseville, et al. [1] who also used a line-by-line scan and a multiple target tracking approach. Two notable exceptions, however, are that the method of [1] did not attempt to form closed contours, whereas DTA contains a track linking strategy for that purpose, and that DTA integrates intensity information as an additional state dimension in a

multidimensional tracking approach whereas [1] used only a single spatial dimension for tracking.

DTA is fully automatic and capable of forming closed contours, even for nonconvex shapes, given the condition where the endpoints of the tracks to be joined each fall within the association gate of the other. This approach is reasonable for synthetic images with consistent edges but in real images, where edge structure is much more complex, DTA forms many contour fragments whose endpoints may not be adequately aligned for the track-linking step to be successful. Another problem of the line-by-line scan is the time analogy since a track can conceivably progress horizontally along an image line, that is, perpendicular to the time step, requiring the acquisition of multiple data points within a single time unit. This time-dimension problem increases when the propagation of two tracks toward each other along a single line is considered.

These problems are sufficiently large that the fully-automatic DTA method was abandoned in favour of a semiautomatic approach where contours are formed beginning from a user-defined starting point. The time dimension is defined to be independent of the spatial dimensions by assuming a unit time interval between all successive edge pixels identified by the tracing process. This approach attempts to form the boundary of a single object and thus allows for easier examination of results and facilitates algorithm refinement.

Method DTB represents this phase of development. Intensity information is obtained by sampling the intensity on each side of the edge rather than at the edge pixel, as was the case for DTA, producing intensity measurements with better stability in real images. Local information, in the form of the distances of candidate pixels to the predicted

position and to the previous position estimate, is used to select the edge pixels for track propagation. However, the use of only local information produces a lack of robustness where a track can easily depart from the desired edge when affected by noise and nearby edges.

Still, the DTB method shows advantages when compared to classical snakes and this is significant. Both methods are 2D in nature, both are edge-based, and both depend primarily on local information for boundary formation. Therefore, both appear to be in a comparable stage of development. After initial publication, the classical snakes and related algorithms enjoyed over a decade of intense research by many groups worldwide, including the identification of major problems, the development of strategies to overcome those problems, and the development of 3D models. At about the same time, edge tracing was receiving less and less attention as an approach to segmentation; however, this appears to be a loss given the advantages in edge selectivity, entry into narrow concavities, execution efficiency, and the influence of parameter selection on final contour shape, shown by the DTB algorithm.

The next step in development, DTC, incorporates several modifications. Spatial edge points found by edge detection are interpolated to obtain subpixel resolution. This, along with the implementation of a Kalman smoother, as suggested in chapter 4, produces contours that are smooth and also capable of forming sharp transitions, highly desirable properties. The most important change, however, is the recognition that the next-point selection in the data association process is actually an optimization problem and that global image information can be introduced within this context. The distance measure is modified, normalized, and combined with the normalized local gradient strength and with

a threshold-based, operator-selected, pixel classification, also normalized to values between 0 and 1. The product of these three elements forms the objective function with which candidate data points are evaluated. This arrangement vastly improves the robustness, permitting the results visible in chapter 5.

In addition, the gradient threshold that was employed in DTB is removed in DTC. Gradient thresholds are typically used in edge-based methods as a way to separate noise edges from valid edges. They can cause difficulties, however, for low-contrast edges in cases where the gradient of the desired edge fluctuates, causing gaps to occur between edge points when the threshold is applied. With DTB, the gradient threshold is required to limit the likelihood of the track being diverted from the desired edge due to the noise. Chapter 4, figure 4-5 shows an example where DTB has been able to jump across a threshold-induced gap; however, the threshold setting is a sensitive parameter with significant influence on the result since points on both sides of the gap must fall within the association gate or the jump cannot be completed. The DTC algorithm is able to operate without any requirement for the user to set a gradient threshold due to the inclusion of gradient intensity and pixel classification in the data association objective function. These significantly reduce the likelihood of track diversion due to noise and, since no threshold is applied, all edge information is retained and is available for use by the track.

The DTC pixel classification can be formed independently of the edge information. This can be accomplished, for example, by applying thresholds to a smoothed version of the image, or by using the classification output from another algorithm. The

independently-determined edge and classification features are then combined by the tracking process.

The DTC methods are flexible and adaptable. Statistically-based features can be tracked as state variables and the data association objective function can be modified to include the effect of other elements, especially those which may not have a statistical foundation. This flexibility encourages further advancement.

Within the framework suggested in chapter 2, DTC can be classed as a second-generation edge tracing algorithm due to the use of the statistically-based Kalman filter. When knowledge, possibly in the form of a fuzzy rule base, is included in the data association process, a third-generation edge tracer would result.

The DTC method can be used to processes 3D images as sets of 2D slices. A 3D extension to the algorithm is highly desirable; however, it remains possible to stack 2D contours to form 3D surfaces, as is done with manually defined contours and as has been done with other algorithms [2].

Remaining problems include dependence on track direction, dependence on starting point, the possibility of self intersection, the possibility that a closed contour is not formed in all cases, and the selection of appropriate parameter values, where several trial runs may be required. Because of these, the DTC method is presently best applied within the context of a user interface that permits the operator to perform automatic tracking and to manually cut and link tracks to form the desired contours. This, however, can still remove some of the tedium of manual tracing while allowing a high-quality result to be achieved. Further research may allow the remaining problems to be overcome.

6.2 Head Model Formation

During the course of the development of these studies, a set of surfaces were extracted from a 3D, MR image of an epilepsy patient. This was done with the DTC algorithm although prior to the integration of the pixel classification feature. The surface extraction required approximately three weeks of effort on the part of an operator who was trained for the task but not a medical expert. The operator expressed the opinion that despite the considerable amount of manual interaction required to start, trim, and link tracks to form contours, use of the edge tracing algorithm reduced operator tedium when compared to manual contour formation, alone.

Figure 6-1 shows a cut-away view of the skin surface, the outer skull surface, and the brain surface after the segmentation of multiple consecutive, transverse, image slices was rendered by the amira™ [3] software visualization tool. The 3D surfaces were obtained by stacking sets of 2D contours. Excellent detail is evident in these surfaces, suggesting that this method will be suitable for use in the formation of electrical head models for epilepsy research. Although the brain surface does show topological inconsistencies in a few locations, particularly gyri that are bridged across intervening sulci, these can be corrected with further manual editing.

In other tests where the pixel classification feature of DTC was included, track lengths improved and the number of tracks required to form complex, convoluted contours was reduced. This suggests that less operator involvement is required when the full DTC algorithm is applied and this should reduce the overall time required to form the segmentation.

Figure 6-1. MR Segmentation Surfaces

6.3 The Medical Image Segmentation Problem

Given the number of algorithms that have been developed and the fact that automatic medical image segmentation remains an unsolved problem, one may wonder if a fully–automatic solution with accuracy matching a human expert will ever be found. It is clear that first-generation methods were not successful and it becomes more and more evident that second-generation methods, despite obvious improvements and mathematical superiority, will likewise prove to be insufficient. The second-generation methods have

produced fully-automatic segmentation systems, such as the BET/FAST combination used in chapter 5; however, obvious misclassification may still occur.

Among the third-generation algorithms, certain atlas-based methods have been shown to produce automatic segmentations that are competitive with manual segmentations [4]-[6], when an overlap metric (e.g. the ratio of the intersection of two segmentations to their union) is used for the comparison. There is indication, though, that such favourable comparison may only occur under certain conditions [6]. Factors that can affect algorithm performance include: 1) the image acquisition parameters; 2) the method for registering the atlas with the test image where registration methods capable of greater deformation produce better comparison results; 3) the manual tracing protocol used by the investigator, which, when identifying normal brain structures, must match that used in the atlas formation, for best comparison; 4) demographics and disease where differences in brain structure between the test subjects and the subjects included in atlas formation may cause significant errors.

In addition, the overlap metric may not adequately identify errors in cases where discrepancies between the two segmentations under examination have relatively small volume. These errors could still be spatially significant, for example, in the case of a narrow projection occurring in only one of the segmentations. The overlap metric will not detect when the automatic segmentation has moved beyond the spatial limits of the inter- and intra-expert variability and, therefore, outside the set of acceptable segmentations, as it has been defined here. Thus, it is not possible to conclude that automatic methods have attained equivalence with human experts.

6.3.1 Knowledge

Production of segmentations that consistently fall within the spatial limits of the inter- and intra-expert variability may require that automatic methods incorporate input from all of the sources that are used by human experts. The nature of those sources may be somewhat difficult to assess since human experts typically require between two and three decades of diverse training, from childhood, before several years of specific training can be applied and the necessary level of expertise achieved. It may be suggested, though, that the knowledge used by an expert incorporates a model of the expected result, including approximate spatial distribution of tissue classes. In addition, explicit recognition of all tissue classes contained within the image appears to occur. The boundary that a human expert would draw, therefore, is influenced by the local intensity variation, the proximity to other tissue classes, the expected spatial relationships between tissue classes, the location within the image, the imaging modality, as well as, possibly, some unknown processing layers. This clearly involves the integration of knowledge from several sources and is well beyond the second-generation algorithms which are typically implementations of optimization methods applied to image data without consideration for the specific content of the image. Results from second-generation methods may, however, prove to be useful precursors.

6.3.2 The Segmentation Standard

The standard for in vivo image segmentation is the human medical expert and since variability occurs among experts [7], the standard is not unique. It has been argued that inter-expert variability may improve with standardized training programs [8] but, even if

successful, this would not remove the intra-expert variability. For a given image, the true segmentation is therefore not a single result but is better described as a random quantity with a distribution derived from the aggregate of all expert segmentations of the image. This distribution may well be multimodal, for example, in cases where the expert segmentations follow spatially distinct paths for portions of the object boundary, producing multiple segmentation interpretations.

6.3.3 Operator Interaction

Automatic segmentation methods have been pursued to overcome the operator tedium that occurs when manual segmentation of large datasets is considered and also to overcome the operator variability that produces poor reproducibility and thus low confidence in manual segmentation results [7]. Partial success can be claimed because automatic segmentation algorithms can often reduce the need for manual intervention. However, automatic results are not always consistent with anatomical realities and in cases where an automatic segmentation result is deemed to be inadequate, a tedious, manual correction may be required. The problem of operator variability also remains since the standard for validating segmentation results is, itself, variable. It is unlikely that automatically generated segmentations will meet with acceptance by all medical experts, on all occasions. A more effective means for reducing operator variability may involve the development of segmentation guidelines and the implementation of standardized training within the medical community, as is being suggested, for example, in the context of radiation therapy [9].

Continued development and refinement of automatic segmentation algorithms should permit further reduction of operator interaction by providing improved solutions. The

development of algorithms that provide multiple segmentation interpretations may also produce a benefit, allowing the operator to choose the interpretation that best matches the desired result, reducing the amount of manual modification required.

6.4 The Role of Edge Tracing in Segmentation

Automatic, recursive edge tracing appears to have been largely abandoned, as a segmentation method, by the medical image analysis community sometime in the early 1990's. The main complaint was poor robustness due to noise sensitivity. The studies presented here identify related problems, including dependence on track direction, dependence on starting point, the possibility of self intersection, and the possibility that a closed contour will not be formed. However, it is also shown that: i) when closed contours can be formed, edge tracing can produce comparable segmentations when compared to other methods over a range of noise, intensity nonuniformity, and partial volume averaging; ii) that edge tracing extracts information not acquired by other methods; and iii) that edge tracing based on target tracking algorithms can combine features from many sources with potential yet to be explored, for example, the inclusion of domain knowledge to guide the tracking process.

The problem of sensitivity to noise indicates that edge tracing should not be applied using local information only. When properly guided by global information, the same sensitivity property gives an edge tracing method the capability of capturing local detail. Acquiring local detail and combining global information are the main advantages of target-tracking–based, edge tracing algorithms. These are significant advantages that may ultimately allow edge tracing algorithms to find greater application in the context of medical image segmentation.

Given that a particular object in an image can be represented by its volume or its surface, either may be used to form the segmentation. By far, the most common approach has been to define the object volume. Even some boundary-forming methods, such as active surfaces, propagate through the object volume rather than along its surface. The processing time for methods that examine the volume data will be much higher than for those that process data near the object surface, assuming similar computational complexity. Therefore, methods that propagate along the object surface, as do edge tracing algorithms, should have an advantage in speed.

Assuming that the segmentation standard is best described by a statistical distribution formed from human expert segmentation results, as has been suggested here, then algorithms capable of estimating a distribution, as are Kalman filters, are capable of a greater degree of information extraction. Furthermore, algorithms exist within the target tracking literature that generate and process multiple hypotheses. Thus, the use of target tracking algorithms in edge tracing may permit the development of automatically–identified, multiple segmentation interpretations. These two items, a statistically-based boundary and multiple interpretations, may ultimately prove to be important components of a segmentation result.

6.5 Conclusions

Presently, automatic medical image segmentation remains an unsolved problem. A great variety of methods have been applied, several of which have had international attention with ongoing development for over a decade. Still, automatic segmentation equivalent to that from a medical expert has not been achieved to a level that permits general application.

It was hypothesized that the data fusion capability of statistically-based, automatic target tracking algorithms may have advantages that would be useful for edge tracing in medical images, for the purposes of segmentation. This appears to be true. When closed contours can be formed, edge tracing based on target tracking algorithms can produce segmentations comparable to those from other methods over a range of noise, intensity nonuniformity, and partial volume averaging. As well, this approach extracts information not acquired by other methods, particularly visible along low-contrast boundaries. Finally, edge tracing based on target tracking algorithms can utilize features from many sources allowing local and global information, including edge, region intensity, and pixel classification information, to be combined to produce the segmentation.

The potential of target tracking algorithms for image segmentation has not yet been fully explored. Algorithms that generate and process multiple hypotheses exist in the target tracking literature but methods for adapting these algorithms for the purposes of image segmentation remain to be developed. In addition, it may be possible to utilize domain knowledge to improve the tracing result, for example, in the analysis and selection of neighbour points.

Target-tracking-based edge tracing can be used to extract surfaces suitable for head model formation for the purposes of EEG source localization; however, these surfaces are currently formed from a series of 2D contours. A fully-3D, surface-tracking implementation remains to be developed and would be an important step. Other remaining problems include dependence on track direction, dependence on starting point, the possibility of self intersection, the possibility that a closed contour is not formed in all

cases, and the selection of appropriate parameters. Further research may allow these problems to be overcome.

The results from the evaluation of the dynamic edge tracing algorithms support an additional, broader conclusion that suitable edge tracing algorithms can provide value for the purpose of medical image segmentation. Edge tracing algorithms receive very little attention as tools for segmentation but research into these algorithms is important and should continue.

6.6 Future Work

To tackle the remaining problems in the DTC algorithm and to reduce the amount of operator interaction, three ideas stand out as immediate directions for future work. The first would be an investigation into the use of adaptive Kalman filters to automatically identify the noise variance parameters that must currently be set by the operator. This would reduce the required level of operator training and the initialization effort since the noise variances are the parameters in the algorithm whose selection is least intuitive. Adaptive Kalman filters for automatic parameter identification have been studied extensively, e.g. [10]–[12].

To address the problems of sensitivity to initial starting point and to track direction, the edge tracing could be modified to involve multiple starting points with independent track formation beginning from each. All edge points selected by these tracks would initially be considered as candidates for the contour. It may then be possible to determine a final set of points by utilizing dynamic programming [13] within a trellis formed from the candidate edge points. The normalized frequency of selection could be used as the transition probabilities between neighbour points. Dynamic programming would then be

used to select the path of highest probability, forming the object boundary. If successful, this would increase the probability of forming a closed contour, should allow for automatically identifying the case of an unclosed contour, and may also prove to be a mechanism for developing multiple segmentation interpretations, by identifying more than one contour representation from the edge point trellis, for example, the set of "n" best paths through the trellis.

The third idea would be to develop an adaptive thresholding strategy with the intent of removing the user interaction currently required for setting the DTC classification thresholds.

6.7 References

[1] M. Basseville, B. Espiau, J. Gasnier, "Edge detection using sequential methods for change in level – Part I: A sequential edge detection algorithm," *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. ASSP-29, No. 1, 1981, pp. 24-31.

[2] T. McInerney, D. Terzopoulos, "Deformable models in medical image analysis: A survey," *Medical Image Analysis*, Vol. 1, No. 2, 1996, pp. 91-108.

[3] http://www.amiravis.com/

[4] B. Fischl, D.H. Salat, E. Busa, M. Albert, M. Dieterich, C. Haselgrove, A. van der Kouwe, R. Killiany, D. Kennedy, S. Klaveness, A. Montillo, N. Makris, B. Rosen, A. M. Dale, "Whole brain segmentation: Automated labeling of neuroanatomical structures in the human brain," *Neuron*, Vol. 33, 2002, pp. 341-355.

[5] A.P. Zijdenbos, R. Forghani, A.C. Evans, "Automatic "pipeline" analysis of 3-D MRI data for clinical trials: Application to multiple sclerosis," *IEEE Transactions on Medical Imaging*, Vol. 21, No. 10, 2002, pp. 1280-1291.

[6] O.T. Carmichael, H.A. Aizenstein, S.W. Davis, J.T. Becker, P.M. Thompson, C.C. Meltzer, Y. Liu, "Atlas-based hippocampus segmentation in Alzheimer's disease and mild cognitive impairment," *NeuroImage*, Vol. 27, 2005, pp. 979-990.

[7] L.P. Clarke, R.P. Velthuizen, M.A. Camacho, J.J. Heine, M. Vaidyanathan, L.O. Hall, R.W. Thatcher, M.L. Silbiger, "MRI segmentation: Methods and applications," *Magnetic Resonance Imaging*, Vol. 13, No. 3, 1995, pp. 343-368.

[8] K. Gurleyik, E.M. Haacke, "Quantification of errors in volume measurements of the caudate nucleus using magnetic resonance imaging," *Journal of Magnetic Resonance Imaging*, Vol. 15, 2002, pp. 353-363.

[9] C.S. Hamilton, M.A. Ebert, "Volumetric uncertainty in radiotherapy," *Clinical Oncology*, Vol. 17, 2005, pp. 456-464.

[10] R.K. Mehra, "On the identification of variances and adaptive Kalman filtering," *IEEE Transactions on Automatic Control*, Vol. 15, No. 2, 1970, pp. 175-184.

[11] A.H. Mohamed and K.P. Schwarz, "Adaptive Kalman filtering for INS/GPS," *Journal of Geodosy*, Vol. 73, 1999, pp. 193-203.

[12] F. Wang and V. Balakrishnan, "Robust Kalman filters for linear time-varying systems with stochastic parametric uncertainties," *IEEE Transactions on Signal Processing*, Vol. 50, No. 4, 2002, pp. 803-813.

[13] I. Pitas, "Digital Image Processing Algorithms," Prentice-Hall, 1993.

Appendix A

mtrack Software Utility

A.1 Introduction

A software utility known as, *mtrack*, was developed and maintained during the course of this research program and forms the functional interface between an operator and a 3D image to be segmented. This software was written in MATLAB® [1] and consists of several menus to permit selection of a 2D slice, extraction of edge points, initialization of track parameters, track initiation, and results display. A database is formed to store contour and surface results for future access. Interfaces to permit manual line drawing as well as the pruning and linking of contour fragments are also included. This provides a combination of automatic and manual features that allows an operator to perform a complete, semiautomatic segmentation from a given 3D MR or CT image. The utility presently consists of over 5000 lines of Matlab code held within 190 files.

The following sections further describe the menu interfaces that were developed for this utility. Note that the terms *track* and *contour* are used interchangeably although a contour is generally taken to be a closed curve whereas a track may not be closed.

A.2 Main Panel

The utility is invoked from the Matlab command window as:

```
>> mtrack(M)
```

where M is a 3D image stored in a Matlab variable. After this command is executed, the main panel, figure A-1, is displayed allowing the operator access to a number of built-in functions. A selected slice from the image, M, is also displayed in the Image Display window.

A typical usage sequence would be to start the utility, load data from a previous session, if any, select the desired slice, perform edge detection on the selected slice, select one or more starting points, and initiate the tracking operation via the large Track button at the bottom of the panel. A track will be formed for each starting point that has been set.

Subsequently, manual editing of the tracks can be performed where tracks can be pruned, linked, or deleted, and manually drawn contours or contour fragments can be inserted, as desired. Contours can then be assigned labels to group them into surfaces and the surfaces can then be displayed or extracted into a separate data file. Editing is generally performed on a designated, primary track. Any existing track or contour can be selected to be the primary.

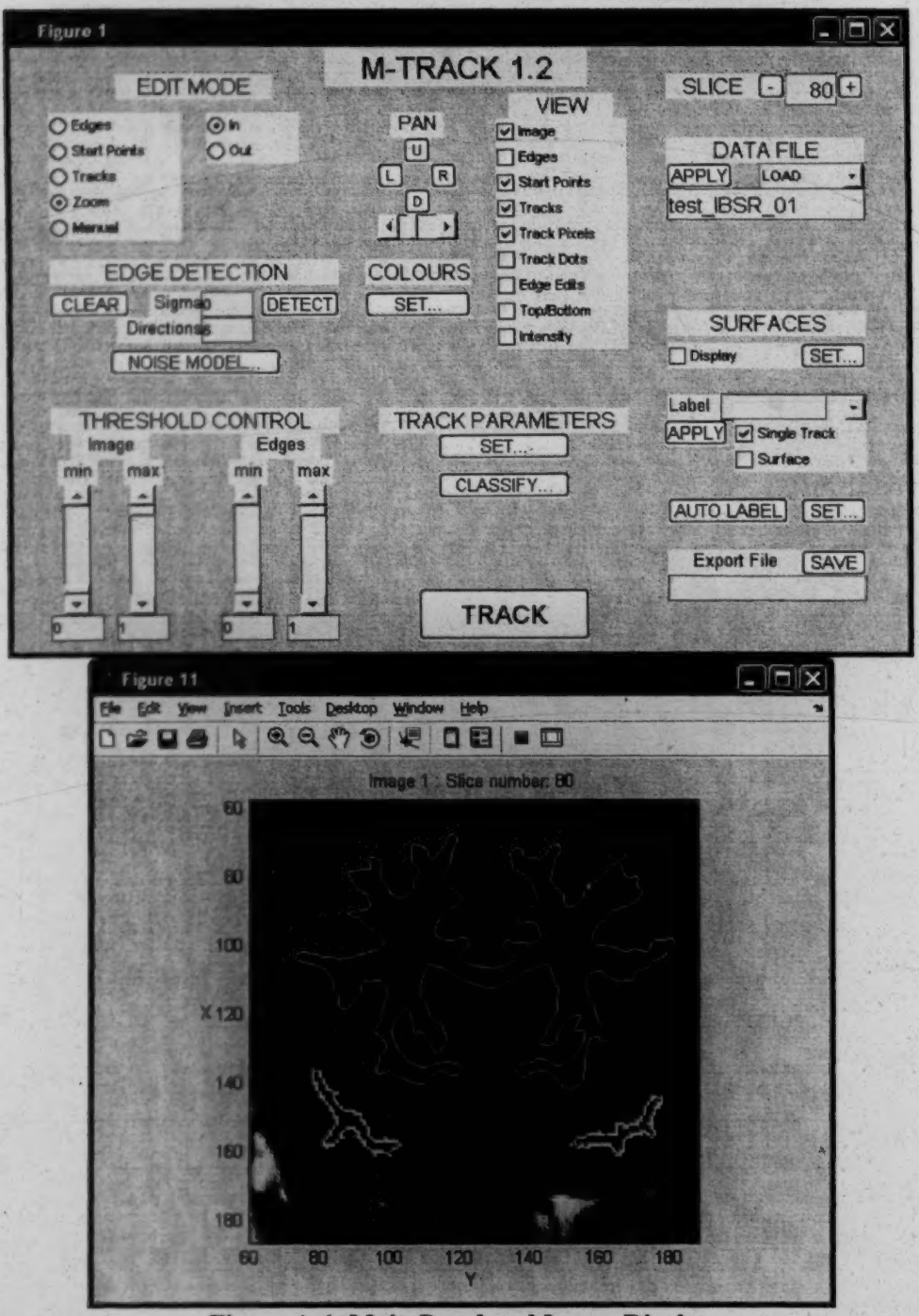

Figure A-1. Main Panel and Image Display

The primary, or selected, track is shown in blue, others in green. Starting points are shown in cyan with a short line between points on each side of the edge.

A.2.1 Slice

One 2D image slice can be displayed. The slice number can be selected by entering the number in the upper right hand corner of the main panel either directly into the display box or using the +/- buttons for unit increment and decrement within the 3D image data.

A.2.2 Data File

The Data File feature in the main panel can be used to load, save, or merge contour data. To activate this feature, the filename is entered, the Load, Save, or Merge mode is selected and the Apply button is pressed. A confirmation dialog box is then employed for action verification. Load will overwrite any existing data, Merge will combine data from the specified file with any existing data, and Save will write existing data to the specified file as a Matlab workspace, suitable for loading through the Load mode at a later time. If the full pathname is not specified in the file name, paths are interpreted relative to the existing Matlab directory setting.

A.2.3 Edit Mode

These selections control the interpretation of mouse activations within the Image Display. There are five primary options in the left column and a set of secondary options to the right. When a primary option is selected, the secondary options will change to match. Each primary option is described below.

A.2.3.1 Edges

The Edges option allows the user to manually remove and restore edge pixels within the edge data. The corresponding secondary options are Delete and Add. Edge detection must first be performed and then the edge pixels can be viewed by selecting View-Edges.

When the Delete secondary option is selected, the left mouse button can be used in the Image Display to delete edge pixels. When the Add secondary option is set, the same mouse activation is used to restore edge pixels. Note that it is not possible to add edge pixels where none exist, the operations are limited to removal and restoration. Locations of deleted edge points can be seen by activating the View-Edge Edits option.

A.2.3.2 Start Points

The Start Points option allows the user to manually add or delete a starting point in preparation for an automatic tracking operation. To add a starting point, the Add secondary option is selected, the cursor is moved to the desired point in the Image Display, and the left mouse button is activated. The strongest edge point within the nearest edge pixel will then be marked as a starting point. To remove a starting point, the Delete secondary option is selected, the cursor is moved to the pixel containing the starting point and the left mouse button is activated. The corresponding starting point will be removed. Note that edge detection must precede start-point selection.

The button labeled All that appears along with the secondary options can be used to set starting points for all visible edge pixels. This option is not typically used, however.

A.2.3.3 Tracks

The Tracks option can be used to manually split, prune, and link tracks into contours. The secondary options are Select, Delete, Split, and Link. These operations can only be performed on existing tracks. A track can be selected as the primary track to highlight it for a subsequent operation by setting the Select option and then left clicking within one of the pixels corresponding to the desired track. Track pixels are made visible by the View-Track Pixels option. The colour of the track pixels will be updated to indicate that

the selected track has become the primary track. Deletion of any track can be performed by selecting the Delete option and then left clicking within one of the track pixels. A track can be split at a desired point by setting the Split option and left clicking within the track pixel closest to the desired point. Also, two tracks can be linked by first selecting one of the two using the Select option, then setting the Link option and clicking within any one of the track pixels of the second track. The closest end points will be joined. The UN button can be used to undo track links and will unlink all components of the primary track.

A.2.3.4 Zoom

The Zoom option can be used to zoom the Image Display. The secondary options are In and Out and control the direction of the zoom. Zooming in is accomplished by setting the In option and left clicking within the Image Display. The image will be zoomed in and the cursor point will be made the centre of the display. Zooming out is accomplished by first setting the Out option and then left clicking within the Image Display.

A.2.3.5 Manual

The Manual option allows the formation and editing of manually defined tracks. A fully-manual segmentation can be performed using this option or, alternatively, manual tracks can be inserted to link two automatic tracks in order to produce closed contours. The secondary options are New and Edit, although New is assumed if no manual tracks currently exist. The New option is used to start a new track and the Edit option is used to begin editing of the primary track. Once one of these options has been chosen, the secondary options become Add, Delete, and End. Selecting the End option will terminate the editing phase. A manual track can be added by selecting the Add option and setting

the cursor at the desired starting point for the track within the Image Display. Pressing and holding the left mouse button followed by movement of the mouse will initiate automatic collection of data points representing a line. Releasing the mouse button stops the automatic collection of data points for the manual track. A single additional point can be added by pressing and releasing the mouse button once or, alternatively, press and hold with subsequent mouse movement for multiple points. The points will be added to the nearest end of the primary track. Individual points can be removed by selecting the Delete option and left clicking on the track pixel to be removed. Note that, if desired, the entire manual track can be removed using the Tracks-Delete option.

A.2.4 Pan

After the zoom feature has been activated, the display window can be adjusted up (U), down (D), left (L), or right (R) through the Pan facility. The slider located directly under these buttons controls the pan step size.

A.2.5 View

The View selections control the display of graphical components in the Image Display. The Image option can be used to display the image, Edges controls display of the edge pixels determined by the edge detection operation, Start Points controls the display of the starting points that have been set, Tracks controls the display of the filtered track, shown as a solid line linking the edge points that are part of the track, Track Pixels controls the display of the pixels containing the edge points for a track, Track Dots controls the display of the subpixel edge points that define a track, Edge Edits controls the display of edge pixels that were removed using Edit Mode-Edges, Top/Bottom controls the display

of a short line between the top-of-slope and bottom-of-slope coordinates on each side of the edge for edge points within the primary track, and Intensity controls the display of the intensity variables, if any, of the primary track on a graph within a separate window.

A.2.6 Threshold Control

Thresholds can be applied to the raw image intensity, or to the edge image. Simple saturation thresholds are used. For the intensity thresholds, image intensity values below the lower threshold are shown as black and intensity values above the upper threshold are shown as white. Intensity values between these limits are scaled for display. If the upper threshold is made less than or equal to the lower threshold, a binary image results. The threshold values displayed in the edit boxes below the sliders are fractions of the maximum intensity level in the selected 2D image.

For the edge thresholds, edge points with gradient levels below the lower threshold or above the upper threshold are removed from consideration. All other edge points are available for use by the tracking system. The threshold values displayed in the edit boxes below the sliders are fractions of the maximum gradient intensity in the selected image.

A.2.7 Edge Detection

Edge detection is performed on the image produced after application of the intensity thresholds of the Threshold Control feature. If no thresholds have been set then the edge detection is performed on the raw image. There are two parameters. Parameter Sigma sets the size of a Gaussian filter used to smooth the image prior to gradient extraction, and Directions sets the number of directions at which the gradient is measured for each pixel. Once these parameters have been selected, edge detection can be initiated by selecting the

Detect pushbutton. The edge data often requires substantial memory resources and this data can be discarded by selecting the Clear pushbutton. This will clear the edge data for the slice that is currently displayed.

A.2.8 Colours

This feature allows the user to set the colours of graphical components of the Image Display as combinations of red, green, and blue levels, each ranging from 0 to 1, as shown in figure A-2. Two predefined colour schemes are available from the Colour Scheme drop-down menu for quickly reverting to default colours.

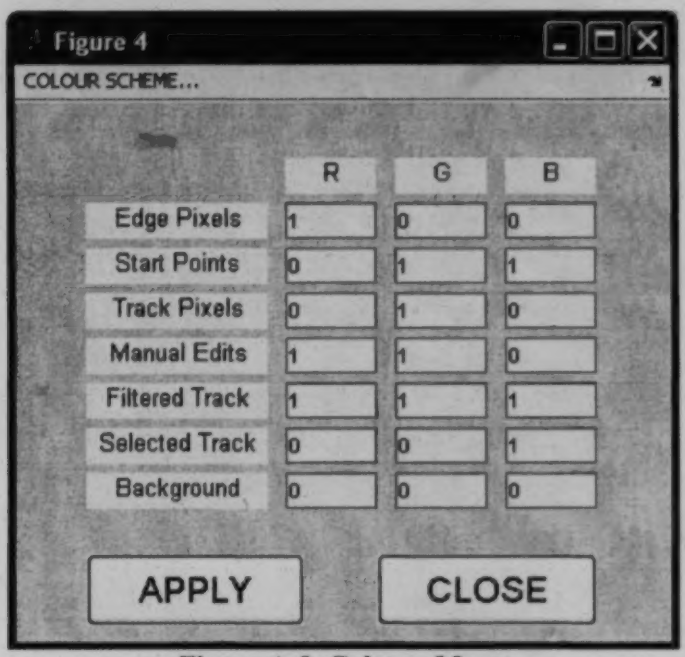

Figure A-2. Colours Menu

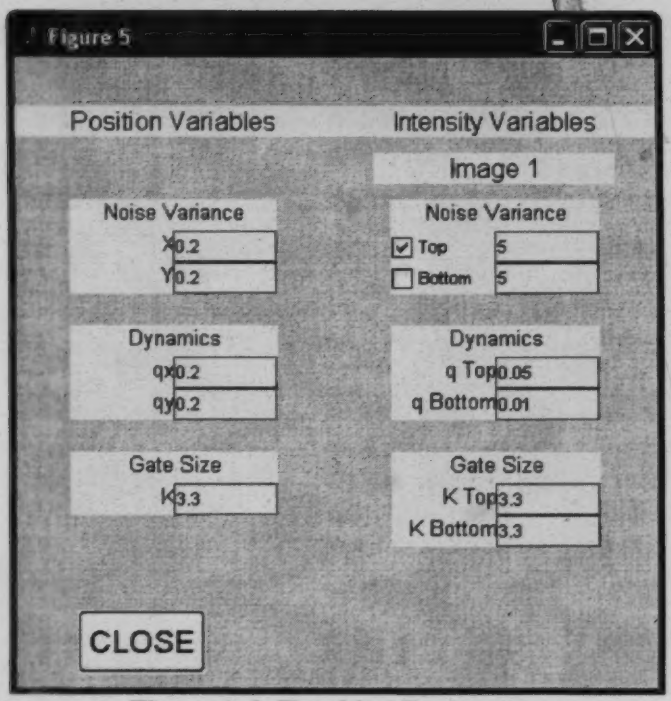

Figure A-3. Tracking Parameters

A.2.9 Track Parameters

A.2.9.1 Set

The measurement noise variances, dynamics parameters, and association gate size constants can be set in this menu, shown in figure A-3, which is displayed when the Track Parameters-Set pushbutton is activated from the Main Panel. The Position Variables correspond to spatial dimensions and Intensity Variables to intensity dimensions in the tracking process. The Top and Bottom intensity variables can be included or excluded by appropriately setting the corresponding checkbox. Track parameters are set and stored on a per track basis. The parameters displayed in the menu are those of the primary track. Parameter settings are captured when the track is formed

so settings can be modified just prior to a tracking operation and these settings will then be applied to the new track.

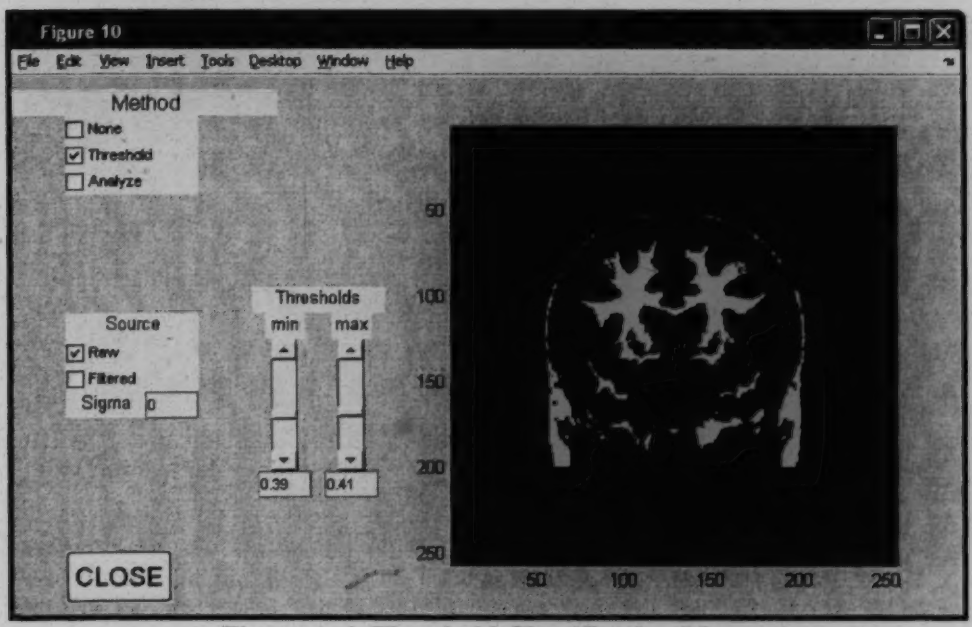

Figure A-4. Threshold Classification Menu

A.2.9.2 Classify

Classification parameters can be set by activating the Track Parameters-Classify pushbutton on the Main Panel which will display the classification menu. Classification can be performed using thresholds, or using a predetermined classification stored in a set of images in files having Analyze 7.5 format. Switching from one to the other can be performed by selecting the appropriate Method option and then specifying the appropriate parameters in the resulting menu. Alternatively, the classification feature can be turned off by selecting the Method-None option. Classification is set on a per track

basis. The menu for Threshold classification is shown in figure A-4 and that for classification data stored in Analyze format files is shown in figure A-5.

Threshold classification is performed by applying two thresholds, an upper threshold and a lower threshold. All pixel intensities below the lower threshold are displayed as dark and all those above the upper threshold are displayed as white. A linear stretch between 0 and 1 is performed for pixel intensities between the two thresholds. This forms a classification image. The numerical threshold values in the edit boxes below the sliders are fractions of the maximum intensity in the image. The source image for the classification can be the raw image or the image filtered by a Gaussian filter with characteristic width specified in pixels.

Classifications from sources like the FAST algorithm used in chapter 5 can be utilized in place of the threshold classification when the classification data are stored in Analyze 7.5 formatted files. These classifications typically correspond to three classes, Gray matter, White matter, and Cerebrospinal fluid. Three filenames can be specified and these are entered without file extensions. Each classification can be used for the Top or the Bottom intensity feature, both, or neither (unused). Thresholds can be applied to further refine the classification images. It is also possible to invert the classification images since the high intensity side of the edge (Top) must be matched with high intensity in the classification image for proper operation.

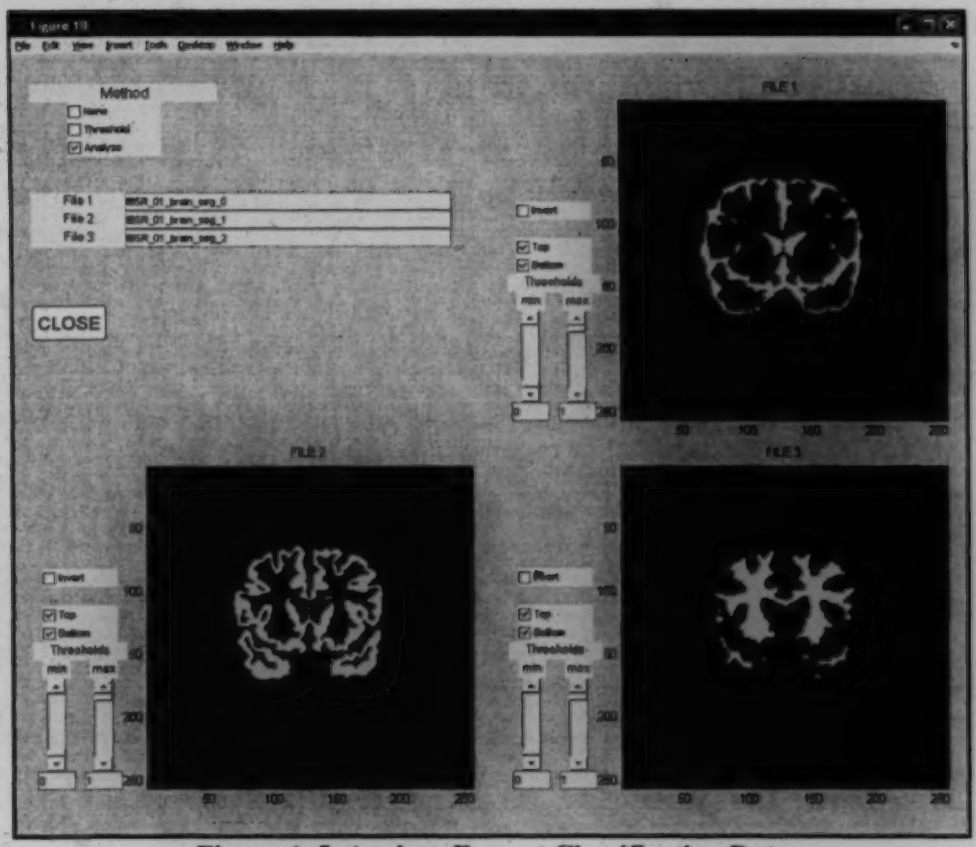

Figure A-5. Analyze Format Classification Data

A.2.10 Surfaces

The surfaces display can be enabled or disabled using the checkbox under the Surfaces section on the main panel. Contours from multiple slices can be organized into surfaces by applying a text label to each contour in the surface. This is done using the Label feature in the Surfaces section of the main panel. The label can be defined as any set of characters and, once used, is available from a drop-down list for easy access. The label can be applied to the primary track in the Image Display by setting the Single Track checkbox prior to activating the Apply pushbutton. Alternatively, an entire surface can be relabeled by selecting the Surface checkbox prior to applying the label change, in which

case the labels of all contours having the same label as that of the primary track will be modified. The surfaces displayed can be exported to a Matlab workspace for off-line processing using the Export feature by specifying a filename and selecting the Save pushbutton.

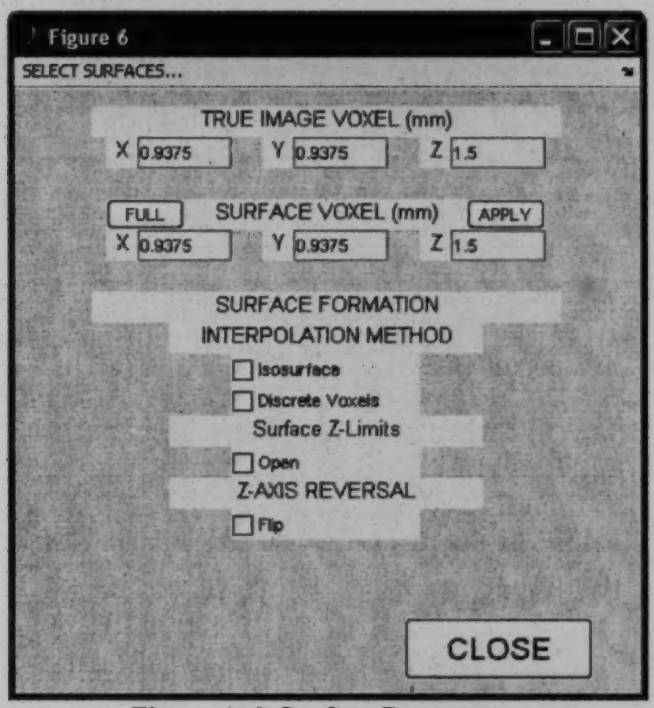

Figure A-6. Surface Parameters

Selecting the Set pushbutton next to the Display checkbox will produce the menu shown in figure A-6. This menu allows the user to set a number of parameters to control the surfaces display. At the top, the voxel size in the image can be set along with the voxel size in the surface display. Adjusting the surface voxel size allows the use of a coarser or finer grid for surface formation, as compared with the image resolution. The voxel dimension orthogonal to the plane of the MR slices cannot be changed.

The surface display is formed by first discretizing all 2D contours in the surface and then displaying these contours as a discretized surface. Two surface interpolation methods are available, Matlab's isosurface produces a smoother display than the Discrete Voxels selection' in which the surface is drawn as a set of binary voxels. Alternatively, the contours can be displayed without surface interpolation by unselecting both of these interpolation methods.

Selecting Open in the Surface Z-Limits section of the menu causes the surface to be displayed such that the contours in the slices at the extremes of the Z dimension are not filled. This gives the effect of an open surface and in some cases may allow visualization of the inside of the surface. Setting the Flip option reverses the order of the slices in the image slice dimension (Z), permitting the surface to be displayed from the opposing viewpoint although it should be recognized that this action also reverses left and right in the displayed surfaces. Example surfaces are shown in figure A-7. The surface can also be zoomed and rotated using the appropriate features in the Matlab toolbar at the top of the display.

Surfaces to be displayed must be selected in the Select Surfaces drop-down menu. This drop-down menu gives a list of all surface labels. Only those surfaces whose labels have been selected are displayed and only those surfaces selected for display will be exported when the Export feature is invoked.

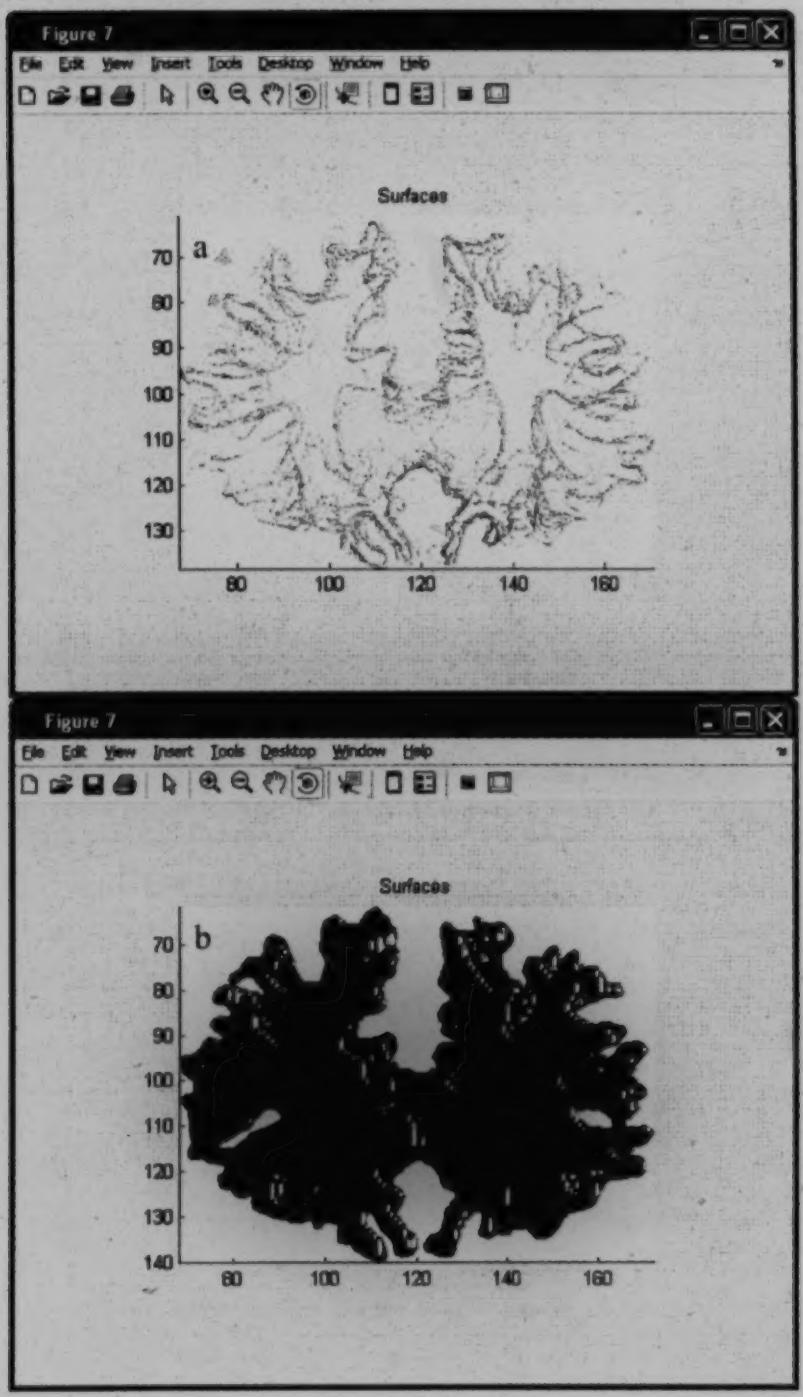
Figure 7
File Edit View Insert Tools Desktop Window Help
Surfaces
a
70
80
90
100
110
120
130
80
100
120
140
160
Figure 7
File Edit View Insert Tools Desktop Window Help
Surfaces
b
70
80
90
100
110
120
130
140
80
100
120
140
160

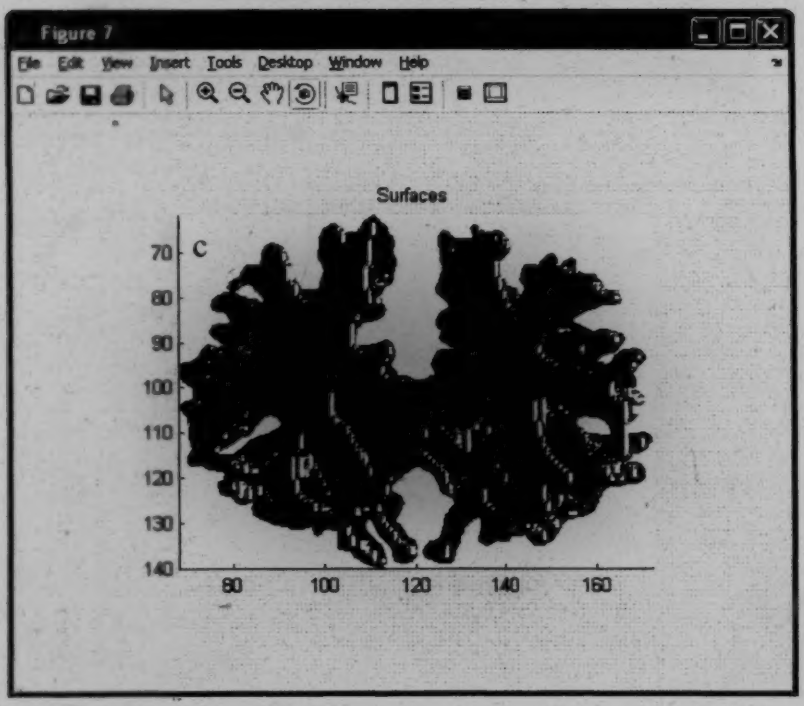

Figure A-7. Surface Display Examples.
Corpus callosum example from chapter 5. (a) contours only; (b) isosurface display; (c) isosurface display with z-axis reversal.

A.3 References

[1] http://www.themathworks.com